

الجمهورية العربية السورية

بسم الله الرحمن الرحيم

" أقرا باسم ربك الذي خلق ، خلق الإنسان من علق ،

أقرا وربك الأكرم الذي علم بالقلم علم الإنسان مالم يعلم "

برامج بلغة سي بلس بلس

C++

الجزء الثاني (لقاءنا في الجزء الثالث)

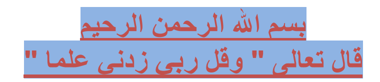
Engineer: Khaled yassin alsheikh

إعداد المهندس خالد ياسين الشيخ

khaledyassinkh@gmail.com

khaledyassinkh@gmail.com

إعداد المهندس خالد ياسين الشيخ



برامج بلغة c++ (الجزء الثاني)

programs in c++ | **جميع outputs من اليسار لليمين**

```
#include<iostream.h>
#include<conio.h>
void p(char * a){
        int i=0;
        while(i<4)
                cout<<a[i++];
        cout<<endl;
}


main(){
        int s=0;
        char x[]="pass" , pass[4];
        for(;;){
                for(int i=0;i<4;i++){
                pass[i]=getch();
                cout<<'*';
                }
                for(i=0;i<4;i++){
                if (pass[i]==x[i]) s++;
                        else
                                break;
        }
        if (s==4){
                cout<<"\npassword correct"<<endl;
                break;
        }
        cout<<endl;
        cout<<"error please enter new password"<<endl;
        s=0;
        }
        p(pass);
        getch();
        return 0;
}
```

الخرج: في حال أدخلنا pass هو:

a. password correct
pass

b. **
password correct
pass**

c. ****
error please enter new password

d. ****
password correct

e. الجواب الصحيح يختلف عما سبق

ملاحظة: التابع getch تقوم بقراءة المحرف دون إظهار المحرف المدخل على الشاشة و التابع موجود في المكتبة conio.h ولذلك يجب تضمين هذه المكتبة قبل استخدام هذا التابع
و في حال أردت إظهار المحرف المدخل على الشاشة استخدم التابع getche()

إعداد المهندس. خالد ياسين الشيخ
khaledyassinkh@gmail.com

الخرج:
يتم طباعة syria arabic بشكل لا نهائي لأن البرنامج يدخل في حلقة لا نهائية. لأن حلقة for لا نهائية.

```
#include<iostream.h>
#include<ctype.h>
void main(){
        while(true){
                for(;;)
                cout<<"syria arabic"<<endl;
                if(1)
                        break;
        }


}
```

بلغ عدد سكان الجمهورية العربية السورية عام 1980 حوالي 12 مليون نسمة و إذا علمت أن معدل تزايد السكان السنوي يبلغ حوالي 2.3% تقريباً و المطلوب كتابة برنامج بلغة c++ يقوم بمايلي:

- إجراء population يقوم بحساب و طباعة عدد سنوات اللازمة لكي يتضاعف عدد السكان حيث يتم طباعة
  - السنة year التي يتضاعف فيها عدد السكان.
  - عدد السنوات.
  - عدد السكان حينها.

الخرج:

year =2011
number of years =31
number of population =24284316.874

```
#include<iostream.h>

#include <iomanip.h>
void population(double n){
        double temp=n;
        long year=1980, i=0;
        while(n<2*temp){
                n=n*1.023;
                year++;
                i++;
        }
        cout<<"year ="<<year<<endl;
                cout<<"number of years
="<<i<<endl;
        cout<<"number of population
="<<setprecision(11)<<n<<endl;
}
void main(){
        double n=12000000;
        population(n);
}
```

اكتب إجرائية frac بلغة c++ يمرر إليها عدد حقيقي بفاصلة و تقوم الإجرائية بطباعته بشكل كسر(بسط / مقام) على نحو **مختزل**
ويراد الحل بدون استخدام أي تابع جاهز.

مثال:
تم تمرير العدد 33.3 يتم طباعة 333/10
تم تمرير العدد 22.4 يتم طباعة 112/5
تم تمرير العدد 0.5 يتم طباعة 1/2
تم تمرير العدد 0.25 يتم طباعة 1/4
تم تمرير 4.125 يتم طباعة 33/8
تم تمرير 50.2 يتم طباعة 251/5

```
#include<iostream.h>
#include<stdio.h>
void f(double & n){
                n=n*2;
}
void frac(double n1){
        double n2,n3,m;
        m=n1;
        int t=n1;
        n2=n1-t;int i=1;
        while(n2!=0){
                i++;
                n1=n1*i;
                t=n1;
                n2=n1-t;
```

<table>
<tr><td>

```
            if(n2!=0)
                  n1=m;
      }
      cout<<n1<<"/"<<i<<endl;;
}
void main(){
      double n1;
      cin>>n1;
      frac(n1);
}
```

</td><td></td></tr>
<tr><td>

```
#include<iostream.h>
#include<conio.h>
int countcall(void){
      static int count=0;
      count++;
      return (count);
}
main(){
      cout<<countcall()<<",";
      cout<<countcall()<<",";
      cout<<countcall()<<endl;
}
```

</td><td>

الخرج:

a. 0,1,2
b. 1,2,3
c.2,3,4
d.3,3,5,6,
e.0,0,0

</td></tr>
<tr><td>

```
#include<iostream.h>
int main(){
      int x=5;
      if(x>5)
            if(x%2==1)
            cout<<"syria";
            else
            cout<<"welcome to syria";
}
```

</td><td>

خرج البرنامج هو:

a. syria
b. welcome to syria
c. syriawelcome to syria
d. لا يتم طباعة أي شيء
e. الجواب الصحيح يختلف عما سبق

</td></tr>
</table>

khaledyassinkh@gmail.com
إعداد المهندس. خالد ياسين الشيخ

<table>
<tr><td>

```
#include<iostream.h>
#include<conio.h>
char *f1(int);
char *f2(int);
char *f1(int generations){
        if(generations==0)
        return ("f1");
        else
                return(f2(generations-1));
}
char *f2(int generations){
        if(generations==0)
                return ("f2");
        else
                return (f1(generations-1));
}
void main(){
        cout<<f1(5)<<endl;
        cout<<f1(5+5)<<endl;
        cout<<f2(10)<<endl;
        cout<<f1(6)<<endl;
}
```

</td><td>

الخرج:

f2
f1
f2
f1

لاحظ عملية الاستدعاءات؟؟

</td></tr>
<tr><td>

```
#include<iostream.h>
#define sqr(n) n*n
main(){
        float x=4.4;
                int m=9;
        cout<<sqr(m)<<endl;

}
```

</td><td>

ما هو خرج البرنامج:

a. **19.36,81**
b. 16.36,81
c. 16.16,19,36
d. 18.32,81
e. الجواب الصحيح يختلف عما سبق

ملاحظة: الماكرو يشبه التابع مع بعض الاختلافات ويمتاز الماكرو عن التابع بالسرعة و سهولة في كتابته و استخدامه أنواع مختلفة من المعطيات دون تحديد المعطيات في ترويسته.

</td></tr>
<tr><td>

```
#include<iostream.h>
#include<math.h>
main(){
        int a=100,b(2),c=20,d=30,e=1,f=128;
        c=sqrt(int(a/b)*3);
        d=fmod(a/pow(b,2),4);
        e=ceil(c/b*0.2);
        f=floor(((a+b+c+d)*2)/2);
        cout<<"c is:   "<<c<<endl;
        cout<<"d is:   "<<d<<endl;
        cout<<"e is:   "<<e<<endl;
        cout<<"f is:   "<<f<<endl;
}
```

</td><td>

الخرج:

c is: 12
d is: 1
d is: 2
f is: 115

</td></tr>
</table>

سبحانك ربنا لا علم لنا إلا ما علمتنا

khaledyassinkh@gmail.com

إعداد المهندس. خالد ياسين الشيخ

Code	الخرج:
#include<iostream.h> main(void){ int s=1; do{ switch(s){ case 1:cout<<"syria"<<endl; case 2: cout<<"welcome to syria"<<endl;break; case 3:cout<<"good luck"<<endl; case 4:cout<<"I'm From syria"<<endl; case 5:cout<<"syria- damascus "<<endl; default:cout<<"information"<<endl; } s++; } while(s<7); }	syria welcome to syria welcome to syria good luck I'm From syria syria- damascus information I'm From syria syria- damascus information syria- damascus information information
#include<iostream.h> #include<conio.h> int x=100; int & f1(){ return x; } void main(){ f1()=x; f1()-=10; x++; cout<<x<<endl; cout<<f1(); getch(); }	91 91
#include<iostream.h> #include<conio.h> int x=100; int & f1(){ return x; } void main(){ f1()=x; f1()-=10; x++; cout<<&x<<endl; cout<<&f1()<<endl; getch(); }	0x00428D50 0x00428D50

khaledyassinkh@gmail.com

إعداد المهندس. خالد ياسين الشيخ

#include<iostream.h> #include<conio.h> int x=100; int *p=&x; int * f1(){ return p; } void main(){ *f1()=2; x++; *p+=1; cout<<x<<endl; cout<<*f1()<<endl; cout<<*p<<endl; getch(); }	الخرج: 4 4 4
#include<iostream.h> #include<conio.h> int x=100; int *p=&x; int * f1(){ return p; } void main(){ *f1()=2; x++; *p+=1; cout<<&x<<endl; cout<<f1()<<endl; cout<<p<<endl; getch(); }	الخرج: 0x00428D50 0x00428D50 0x00428D50
#include<iostream.h> #include<conio.h> int x=100; int *p=&x; int * f1(){ return p; } void main(){ *f1()=2; x++; (*p)++; cout<<&x<<endl; cout<<f1()<<endl; cout<<*p<<endl; cout<<*p<<endl;	الخرج: 0x00428D50 0x00428D50 4 4

<table>
<tr><td>

```
        getch();
}
```

</td><td></td></tr>
<tr><td>

```
#include <iostream.h>
template< class T >
void printArray( const T *array, const int
count )
{
  for ( int i = 0; i < count; i++ )
       cout << array[ i ] << " ";

  cout << endl;

}


int main()
{
  const int aCount = 5, bCount = 7, cCount =
15;

  int a[ aCount ] = { 1, 2, 3, 4, 5 };
  double b[ bCount ] = { 1.1, 2.2, 3.3, 4.4, 5.5,
6.6, 7.7 };
  char c[ cCount ] = "HELLO To SYRIA";
  cout << "Array a contains:" << endl;

  printArray( a, aCount );
  cout << "Array b contains:" << endl;
  printArray( b, bCount );
  cout << "Array c contains:" << endl;
  printArray( c, cCount );
  return 0;
}
```

</td><td>

الخرج:

Array a contains:
1 2 3 4 5
Array b contains:
1.1 2.2 3.3 4.4 5.5 6.6 7.7
Array c contains:
HELLO To SYRIA

</td></tr>
<tr><td>

```
#include<iostream.h>
int main(void){
        int a;
        int *p;
        a=7;
        p=&a;

cout<<"the address of a is"<<&a<<"\n\nthe
contents stored in p is"<<p;
cout<<"\n\nthe stored in a is"<<a<<"\n\nthe
value to which p pointer is"<<*p<<endl;
```

</td><td>

الخرج:

the address of a is0x0012FF7C

the contents stored in p is0x0012FF7C

the stored in a is7

the value to which p pointer is7

و يمكن تمثيل ذلك بالذاكرة الرئيسية كما يلي:

</td></tr>
</table>

إعداد المهندس. خالد ياسين الشيخ

khaledyassinkh@gmail.com

Memory address	Variable a(7)-integer is 4 bytes
0x0012FF7C	00000000 00000000 00000000 00000111

```
return 0;
}
```

الخرج:

Assalamu Alaikum

```
#include<iostream.h>
void main(){
	int i=2,x=5,*p=&x;
	*p=3;
	i=x;
	while(i!=0){
		if(i %2==0)
		cout<<"Assalamu
Alaikum"<<endl;
		i--;
	}
}
```

فائدة(التوابع السطرية) inline function و الفرق بينها وبين function التوابع العادية

لدينا التابع التالي:

```
int sqr(int x){
return (x*x);
}
```

حيث عمل هذا التابع هو أنه يعطينا مربع عدد مدخل و في التابع الرئيسي نستطيع أن نكتب هذا الاستدعاء وفي كل مرة يتم التعريف التابع على أنه inline يقوم المترجم باستبدال التابع بالتابع نفسه.

```
int x=sqr(10);
```

قيمة x هي ١٠*١٠=١٠٠ لنتخيل أننا استخدمنا التابع السابق لتربيع ٢٠٠ رقم في حلقة for مثلا و في كل مرة يصل المترجم إلى تنفيذ التابع يتم استدعاء التابع و تنفيذ ما بداخله ثم إرجاع القيمة إلى المتحول x

و لكن في هذه الحالة نذهب ثم نعود للتابع الرئيسي 200 مثلا وهذا يأخذ الوقت

و الآن لنكتب

```
inline int sqr(int x){
return (x*x);
}
```

يمكنك استدعاء التابع تماماً كما فعلنا في المرة السابقة، و لكن هذه المرة عند

```
#include<iostream.h>
inline Double(int);
void main(){
	int target=6;
	target=Double(target);
	cout<<"Target="<<target<<endl;
	target=Double(target);
	cout<<"Target="<<target<<endl;
	target=Double(target);
	cout<<"Target="<<target<<endl;
}


inline Double(int target){
	int result;
	result=2*target;
}
```

khaledyassinkh@gmail.com

إعداد المهندس خالد ياسين الشيخ

<table>
<tr>
<td>استدعاء التابع سيقوم المترجم خلال عملية الترجمة بالتالي:
int=x*x;
سيتم استبدال التابع ووضع x*x مباشرة و بالتالي لم يعد هناك ذهاب و عودة إلى التابع
الخرج:
Target=12
Target=24
Target=48
ملاحظة هامة:
ينصح استخدام التوابع السطرية فقط مع التوابع الصغيرة .
التوابع السطرية تزيد من سرعة التنفيذ في حال التوابع الصغيرة.</td>
<td></td>
</tr>
<tr>
<td>الخرج:
1 1
2 4
3 9
4 16
5 25
6 36
7 49
8 64
9 81
10 100</td>
<td>

```
#include<iostream.h>
#include<iomanip.h>
inline int sqr(int n){
	return n*n;
}
void main(){
for(int i=1;i<=10;i++)
cout<<i<<setw(4)<<sqr(i)<<endl;


}
```

</td>
</tr>
<tr>
<td>سلسلة فيبوناشي (Fibonacci series):
هي سلسلة من الأرقام حيث أنه أي رقم في هذه السلسلة يساوي مجموع الرقمين السابقين له و يكون الرقم الأول من السلسلة و الثاني منها يساوي الواحد
مثال:
1 1 2 3 5 8 13 21 34 ....n
و المطلوب كتابة برنامج بشكل تكراري يقوم بإيجاد عناصر سلسلة فيبوناشي من أجل n عنصر .</td>
<td>

```
#include<iostream.h>
void Fibonacci(int n){
	int a=1,b=a;
	cout<<a<<" "<<b<<" ";
	for(int i=3;i<=n;i++){
		int s=a+b;
		cout<<s<<" ";
		a=b;
		b=s;
	}
	cout<<endl;
}
void main(){
	int n=9;
	Fibonacci(n);

}
```

</td>
</tr>
</table>

إعداد المهندس. خالد ياسين الشيخ

khaledyassinkh@gmail.com

اكتب بلغة c++ برنامج يقوم بفرز نسق وطباعته.

الخرج:

54321
12345

```
#include<iostream.h>
sort(int v[],int n){
	int i,j;
	for(i=0;i<n;i=i+1)
	for(j=i-1;j>=0&&v[j]>v[j+1];j=j-1)
		{
			int t=v[j];
			v[j]=v[j+1];
			v[j+1]=t;
		}
}
print(int a[],int n){
	int i=0;
	while(i<n)
		cout<<a[i++];
	cout<<endl;
}

void main(){
	int a[5]={5,4,3,2,1};
print(a,5);
sort(a,5);
print(a,5);


}
```

اكتب برنامج يعمل على قراءة n قيمة صحيحة ثم يحسب الوسط الحسابي و الانحراف المعياري لهذه القيم باستخدام الأنساق.

```
#include<iostream.h>
#include<math.h>
int sqr(int n){
	return n*n;}
void main(){
	const int n=5;
	int *mat=new int[n];
	int i,sum=0;
	double val=0,avr,s;
	for(i=0;i<n;i++){
cout<<"enter the value "<<i+1<<endl;
		cin>>mat[i];
		sum+=mat[i];
	}
	avr=(float)sum/n;
	for(i=0;i<n;i++)
		val+=sqr(mat[i]-avr);
	s=sqrt(val/n);
	cout<<"average="<<avr<<endl;
	cout<<"S="<<s<<endl;
	delete[]mat;
}
```

إعداد المهندس خالد ياسين الشيخ
khaledyassinkh@gmail.com

١- اكتب بلغة c++ تابع minimum يعيد أصغر عنصر في المصفوفة(نسق).
٢- اكتب بلغة c++ تابع maximum يعيد أكبر عنصر في المصفوفة(نسق).
الخرج:

```
8.2
1.5
```

```
#include<iostream.h>
#include<conio.h>

double min(double *mat,int n)
{
	int i=0;
	double s=*(mat+i);
	for( i=1;i<n;i++)
	if(mat[i]<s)
		s=mat[i];
		return s;
		return -1;
}

double max(double *mat,int n)
{
	int i=0;
	double s=*(mat+i);
	for( i=1;i<n;i++)
		if(mat[i]>s)
		s=mat[i];
		return s;
		return -1;
}
#define n 10
int main(void){
 double
mat[n]={5,8,6,2,2.5,1.5,7,7.9,8.2,1.9};
	cout<<max(mat,n)<<endl;
	cout<<min(mat,n)<<endl;
	return 0;
}
```

سبحانك ربنا لا علم لنا إلا ما علمتنا

إعداد المهندس خالد ياسين الشيخ
khaledyassinkh@gmail.com

المطلوب:برنامج يقوم بضرب مصفوفتين حيث :
لدينا مصفوفتين ثنائية البعد المصفوفة A تحوي n سطر و m عمود و المصفوفة B تحوي k عامود و m سطر لأن شرط الضرب يجب أن يكون محققاً و بالتالي فإن عدد أعمدة المصفوفة A و أسطر المصفوفة B متساوي.
اكتب إجرائية mult تقوم بعملية الضرب ووضع الناتج في مصفوفة ثالثة C.

```
#include<iostream.h>
#include<iomanip.h>
const int n=10;
        const int m=8;
        const int k=12;
        void mult(int A[][m],int B[][k],int C[][k]){

                for(int i=0;i<n;i++)
                        for(int j=0;j<k;j++){
                                C[i][j]=0;
                                for(int l=0;l<m;l++)

        C[i][j]+=A[i][j]*B[l][j];
                        }
        }
        void read1(int a[][m]){
                for(int i=0;i<n;i++)
                        for(int j=0;j<m;j++){

        cout<<"A["<<i<<"]"<<"["<<j<<"]";
                                cin>>a[i][j];
        }
        }
        void read2(int b[][k]){
                for(int i=0;i<n;i++)
                        for(int j=0;j<k;j++){

        cout<<"B["<<i<<"]"<<"["<<j<<"]";
                                cin>>b[i][j];}
        }
        void print(int C[][k]){
                for(int i=0;i<n;i++){
                        for(int j=0;j<k;j++)

        cout<<setw(4)<<C[i][j]<<setw(4);
                        cout<<endl;
                }
        }


void main(){
        int A[n][m];
        int B[m][k];
        int C[n][k];
        read1(A);
        read2(B);
        mult(A,B,C);
```

<table>
<tr><td>

```
	print(C);
}
```

</td><td></td></tr>
<tr><td>

```
#include<iostream.h>
struct pers{
	char fname[20];
	char  lname[20];
	int age;
};
void insertdata(pers mat[],int n){
	for(int i=0;i<n;i++){
		cin>>mat[i].fname;
		cin>>mat[i].lname;
		cin>>mat[i].age;
}

}
void swap(pers mat[],int i,int j){
	pers t=mat[i];
	mat[i]=mat[j];
	mat[j]=t;
}
void sortdata(pers mat[],int n){
	for(int i=0;i<n-1;i++)
		for(int j=0;j<n-1;j++)

	if(mat[j].fname>mat[j+1].fname)
```

</td><td>اكتب برنامج يعمل من خلال سجل على إدخال بيانات تتعلق بـــ n شخص و يحوي السجل الاسم الاول fname و الاسم الأخير lname و العمر age . و تخزين هذه البيانات في مصفوفة mat ثم يرتب هذه البيانات ترتيباً تصاعدياً و في حال تساوي الاسم الأول يفحص الكنية. ثم يقوم البرنامج بطباعة البيانات قبل وبعد الفرز.</td></tr>
</table>

khaledyassinkh@gmail.com

إعداد المهندس. خالد ياسين الشيخ

```
            swap(mat,j,j+1);
                            else

            if(mat[j].fname==mat[j+1].fname){

            if(mat[j].lname>mat[j+1].lname)

            swap(mat,j,j+1);}
}
void print(pers mat[],int n){
            for(int i=0;i<n;i++)
                        cout<<mat[i].fname<<"
"<<mat[i].lname<<" "<<mat[i].age<<endl;
}


void main(){
            int n;
            cout<<"enter size array ";
            cin>>n;
            pers * mat=new pers[n];
            insertdata(mat,n);
            cout<<"--------"<<endl;
            print(mat,n);
            cout<<"--------"<<endl;
            sortdata(mat,n);
            print(mat,n);
}
```

سبحانك ربنا لا علم لنا إلا ما علمتنا

إعداد المهندس خالد ياسين الشيخ

khaledyassinkh@gmail.com

المطلوب كتابة بلغة C++ برنامج يقوم بتعريف مصفوفة من الأعداد الصحيحة تحوي r سطر و c عمود .

- كتابة إجرائية print لطباعة المصفوفة بطريقة مرتبة (صفوف و أعمدة)

١) كتابة reverserow لطباعة بترتيب عكسي للصفوف

٢) كتابة column لطباعة كل عمود بسطر.

٣) كتابة إجرائية sumrow لطباعة مجموع كل صف.

٤) كتابة إجرائية sumcolumn لطباعة مجموع كل عمود.

٥) كتابة إجرائية maxcolumn لطباعة العنصر الأكبر من كل عمود

٦) كتابة إجرائية maxrow لطباعة العنصر الأكبر من كل صف

٧) تابع summat يعيد مجموع عناصر المصفوفة ككل.

```
#include<iostream.h>
#include<iomanip.h>
const int r=3,c=4;
void print(int a[][c]){
for(int i=0;i<r;i++){
for(int j=0;j<c;j++)
cout<<a[i][j]<<setw(4);
cout<<endl;
}
}
void reverserow(int a[][c]){
        for(int i=0;i<r;i++){
                for(int j=c-1;j>=0;j--)

        cout<<a[i][j]<<setw(4);
                cout<<endl;
        }
}
void column(int a[][c]){
        for(int j=0;j<c;j++){
                for(int i=0;i<r;i++)

        cout<<a[i][j]<<setw(4);
                cout<<endl;
        }
}
void sumrow(int a[][c]){
        int sum;
        for(int i=0;i<r;i++){
                sum=0;
                for(int j=0;j<c;j++)
                        sum+=a[i][j];

        cout<<"row"<<i+1<<"="<<sum;
                cout<<endl;

        }
}
void sumcolumn(int a[][c]){
        int sum;
        for(int j=0;j<c;j++){
                sum=0;
                for(int i=0;i<r;i++)
                        sum+=a[i][j];

        cout<<"column"<<j+1<<"="<<sum;
                        cout<<endl;
        }
}
```

khaledyassinkh@gmail.com

إعداد المهندس. خالد ياسين الشيخ

```
void maxcolumn(int a[][c]){
        int max;
        for(int j=0;j<c;j++){
                max=a[0][j];
                for(int i=1;i<r;i++)
                        if (a[i][j]>max)
                                max=a[i][j];


        cout<<"maxcolumn"<<j+1<<"="<<max
;
                        cout<<endl;
        }
}
void maxrow(int a[][c]){
        int max;
        for(int i=0;i<r;i++){
                max=a[i][0];
                for(int j=1;j<c;j++)
                        if(a[i][j]>max)
                                max=a[i][j];

        cout<<"maxrow"<<i+1<<"="<<max;
                        cout<<endl;
        }
}
int summat(int a[][c]){
        int sum=0;
        for(int i=0;i<r;i++)
                for(int j=0;j<c;j++)
                        sum+=a[i][j];
                return sum;
}


void main(){
int a[r][c]={1,2,3,4,
        5,6,7,8,
        9,10,11,12};
print(a);
cout<<endl;
reverserow(a);
cout<<endl;
column(a);
sumrow(a);
sumcolumn (a);
cout<<endl;
maxcolumn(a);
cout<<endl;
```

khaledyassinkh@gmail.com

إعداد المهندس. خالد ياسين الشيخ

<table>
<tr><td>

```
maxrow(a);
cout<<endl;
cout<<"sum items all
array="<<summat(a)<<endl;

}
```

</td><td></td></tr>
<tr><td>

```
#include<iostream.h>
#include<iomanip.h>
void main(){
        int j,s=0;
        int mat[][3]={1,2,3,
                4,5,6,
                7,8,9};
                for(int i=0;i<3;i++)
        for( j=0;j<3;j++)
                if(i==0||i==2 )
                cout<<mat[i][j]<<" ";



}
```

</td><td>خرج البرنامج هو:
1 2 3 7 8 9</td></tr>
<tr><td>

```
#include<iostream.h>
char * f1(char *syria,int n,int a[]){
        cout<<*syria<<endl;
        cout<<syria<<endl;
        cout<<syria[a[n]]<<endl;
        return syria;
}

void main(){
        char *s="arabic";
        int a[9]={1,3,4};
        s=f1(s,4,a);
        cout<<s<<endl;
}
```

</td><td>الخرج:
a
arabic
a
arabic</td></tr>
<tr><td>

```
#include<iostream.h>
int max=4;
int *p=&max;
int &f1(int n){
        max=max+n/2*3+4;
        return max;
}
int *f2(int n){
        *p=*p+n/2+4;
        return p;
}
void main(){
        f1(max);
        f2(max);
```

</td><td>الخرج:
25
25</td></tr>
</table>

```
		cout<<max<<endl;
		cout<<*p<<endl;
}
```

المطلوب:كتابة برنامج يعمل على قراءة مصفوفة مربعة يحدد بعدها المستخدم ثم يحسب هذا البرنامج:

١- مجموع عناصر كل سطر(تابع p1).
٢- مجموع عناصر كل عامود(تابع p2).
٣- مجموع كل العناصر(تابع p3).
٤- مجموع عناصر القطر الرئيسي(تابع p4).
٥- مجموع عناصر القطر الثانوي(تابع p5).
٦- إجرائيات الطباعة المناسبة.

ملاحظة: المطلوب تحقيق جميع ما سبق باستخدام المؤشرات حصراً.

```
#include<iostream.h>
#include<iomanip.h>
void read(int *a,int n){
		for(int i=0;i<n;i++)
			for(int j=0;j<n;j++){

		cout<<"["<<i<<"]"<<"["<<j<<"]";
				cin>>*(a+n*i+j);}
}
void print(int *a,int n){
		for(int i=0;i<n;i++)
		{
			for(int j=0;j<n;j++)

		cout<<*(a+n*i+j)<<setw(4);
			cout<<endl;
		}
}
void printarrangement(int suml[],int
sumc[],int n){
		for(int i=0;i<n;i++){

		cout<<"line"<<i+1<<"summery="<<su
ml[i]<<endl;

		cout<<"column"<<i+1<<"summery="<
<sumc[i]<<endl;
		}

}
void p1(int *a,int n,int suml[]){
		for(int i=0;i<n;i++)
		{
			suml[i]=0;
			for(int j=0;j<n;j++)
				suml[i]+=*(a+n*i+j);
		}


}
void p2(int*a,int n,int sumc[]){
		for(int j=0;j<n;j++){
			sumc[j]=0;
		for(int i=0;i<n;i++)
```

khaledyassinkh@gmail.com

إعداد المهندس. خالد ياسين الشيخ

```
                    sumc[j]+=*(a+n*i+j);
}}
int p3(int suml[],int n ){
          int sumt=0;
          for(int i=0;i<n;i++)
                    sumt+=suml[i];
          return sumt;
}
int p4(int *a,int n){
          int sumd=0;
          for(int i=0;i<n;i++)
                    sumd+=*(a+n*i+i);
          return sumd;
}
int p5(int *a,int n){
          int sumd=0;
          for(int i=n;i>0;i--)
                    sumd+=*(a+n*(n-i)+(i-1));
          return sumd;
}

void main(){
          cout<<"enter "<<endl;
          int n;
          cin>>n;
          int *suml=new int[n];
          int* sumc=new int[n];
          int **a;
           a=new int *[n];
           for(int i=0;i<n;i++)
                     a[i]=new int [n];
           read(&a[0][0],n);
           p1(&a[0][0],n,suml);
           p2(&a[0][0],n,sumc);
           p3(suml,n);
           print(&a[0][0],n);
           cout<<p3(suml,n)<<endl;
           printarrangement(suml,sumc,n);
           cout<<p4(&a[0][0],n)<<endl;
           cout<<p5(&a[0][0],n)<<endl;
delete []sumc;
           delete []suml;
           for (i=0;i<n;i++)
                     delete []a[i];
           delete []a;

}
```

إعداد المهندس خالد ياسين الشيخ

khaledyassinkh@gmail.com

اكتب برنامج بلغة ++c يعرف نسق يحوي n عنصر و المطلوب:
1 – إجراء لتعبئة عناصر النسق.
2- إجراء يضع في كل خانة مجموع الخانة نفسها مع الخانة التي تسبقها.
3- إجراء لطباعة عناصر النسق.

ملاحظة هامة: ما الفرق بين الإجراء sumprev و الإجراء sum.

```
#include<iostream.h>
void readmat(int *mat,int n){
	for(int i=0;i<n;i++)
	{
		cout<<"mat["<<i<<"]= ";
		cin>>*(mat+i);
	}
}
void print(int mat[],int n){
	int i=0;
	for(;i<n;i++)
		cout<<*(mat+i)<<",";
	cout<<endl;
}
void sumprev(int mat[],int n){
	for(int i=1;i<n;i++)
		mat[i]+=mat[i-1];
}
void sum(int mat[],int temp[],int n){
	for(int i=0;i<n;i++)
	temp[i]=mat[i];
	for(i=1;i<n;i++)
		mat[i]+=temp[i-1];
}

void main(){
	const n=5;
	int mat[n],temp[n];
	readmat(mat,n);
	print(mat,n);
	sum(mat,temp,n);
	print(mat,n);
	sumprev(temp,n);
	print(temp,n);
}
```

اكتب إجراء يقوم بحساب وطباعة قواسم عدد ما عدا العدد نفسه عن طريق **نسق** حيث يطلب تخزين هذه القواسم في نسق ثم طباعة القواسم على شاشة الحاسب.
مثال: قواسم العدد 6 هي ,1,2,3
قواسم العدد 18 هي ,1,2,3,6,9
قواسم العدد 13 هي 1
قواسم العدد 100 هي 1,2,4,5,10,20,25,50

```
#include<iostream.h>
void divider(int mat[],int n,int &count){
count=0;
for(int i=1;i<=n/2;i++)
if(n%i==0){
mat[count]=i;
count++;
}
}
void print(int mat[],int n){
	int i=0;
	for(;i<n;i++)
		cout<<*(mat+i)<<",";
	cout<<endl;
```

khaledyassinkh@gmail.com

إعداد المهندس. خالد ياسين الشيخ

```
}

void main(){
	const max=200;
	int mat[max];int c,n;
	cin>>n;
	divider(mat,n,c);
	print(mat,c);
}
```

اكتب برنامجاً يقرأ عناصر مصفوفة مربعة N*N ثم يختبرها إذا كانت ذات قطر مسيطر أم لا.

ملاحظة : تكون المصفوفة ذات قطر مسيطر إذا كانت القيمة المطلقة لكل عنصر في القطر الرئيسي أكبر من مجموع القيم المطلقة للعناصر الموجودة معه في نفس السطر.

المصفوفة التالية ذات قطر مسيطر:

$$\begin{bmatrix} 3 & 0 & 1 \\ 1 & 5 & 3 \\ 2 & 2 & 5 \end{bmatrix}$$

المصفوفة التالية ذات قطر مسيطر

$$\begin{bmatrix} -3 & 0 & -1 \\ 1 & 5 & 3 \\ 2 & 2 & -5 \end{bmatrix}$$

```
#include<iostream.h>
#include<math.h>
void readmat(int *mat,int n){
			for (int i=0; i<n;i++)
				for (int j=0; j<n;j++){

cout<<"mat["<<i<<"]"<<"["<<j<<"]";
  cin>>*(mat+i*n+j);
				}
}
bool dominant(int *mat,int n){
	int Sum =0;
 for ( int i=0; i < n; i++) {
  for (int j=0; j < n; j++)
   if (i != j)
    Sum += abs(*(mat+i*n+j));
  if (abs(*(mat+i*n+i))<=Sum)
		return false;
  Sum = 0;
 }
 return 1;
}
void print(int *mat,int n){
	for(int i=0;i<n;i++){
		for(int j=0;j<n;j++)
		cout<<*(mat+i*n+j)<<" ";
		cout<<endl;
}
}
void main(){
	const  n=3;
	int mat[n][n];
	readmat(&mat[0][0],n);
	print(&mat[0][0],n);
	if((dominant(&mat[0][0],n))==true)
		cout<<"The Array is
dominant" <<endl;
		else
		cout <<"The Array is not
dominant" << endl;
```

إعداد المهندس خالد ياسين الشيخ
khaledyassinkh@gmail.com

<table>
<tr><td>

```
}
```

</td><td></td></tr>
<tr><td>

```
#include <iostream.h>
const int max = 400;
void main()
{
 int y, large = 0, small = 0, number =0, other=0;
 char Mat[max];
 cin >> Mat[0];
 int i=0;
 while ( (Mat[i] != '.') && (i < max) ) {
  y = Mat[i];
  if ( (y >= 65) && (y <= 90) )
   large++;
  else if ( (y >= 97) && (y <= 122) )
   small++;
  else
  if (y>='0' &&y<='9')
              number++;
  else
              other++;

  cin >> Mat[++i];
 }
 cout << "large count = " << large<< endl;
 cout << "small count = " << small << endl;
 cout << "number count = " << number <<
endl;
 cout << "other count = " << other << endl;
}
```

</td><td>أكتب برنامج يقرأ نص باللغة الإنكليزية و يحسب طول النص علماً أن النص ينتهي بـــنقطة (.) ثم يحسب عدد كل من الأحرف الصغيرة و الكبيرة و الأوقام و الرموز الاخرى...
ملاحظة : سنرمز للحروف الكبيرة بـ ـــ large
و نرمز للحروف الصغيرة بــ ـــــ small
و نرمز للأرقام بــ ـــــ number
و نرمز للرموز الأخرى بــ ـــــ other</td></tr>
<tr><td>

```
#include<iostream.h>
void main(){
char mat[400];
int i=0;
cin>>mat[0];
while(mat[i]!='.'){
            cin>>mat[++i];
}
int s=i;
for(int j=i-1;j>=0;j--)
cout<<mat[j];
cout<<endl;
}
```

</td><td>اكتب برنامج بلغة c++ يقرأ سلسلة من الحروف ثم يقوم بطباعتها بشكل معكوس علما أن السلسلة تنتهي بـــنقطة (.).
مثال تم إدخال .yassin تطبع على الشاشة nissay
تم إدخال .omar ali تطبع على الشاشة ialramo</td></tr>
</table>

إعداد المهندس خالد ياسين الشيخ

khaledyassinkh@gmail.com

اكتب بلغة c++ تابع findroot لحساب القيمة التقريبية لجذر تربيعي لعدد حقيقي عندمايصبح القيمة التقريبية للجذر هي 1%.
مثال value=4 يرد التابع 1.99939
value=100 يرد التابع 9.96753
Value=15 يرد التابع 3.87133
Value=100.9 يرد التابع 10.0113 و هكذا دواليك.........

```
#include<iostream.h>
float findroot(float value){
	float testval=1,
		res1,
		deviation,
		normdev;
	do{
		res1=value/testval;
		deviation=res1-testval;
		testval=testval+(deviation/2);
		normdev=(deviation/res1);
	}
	while((normdev>0.01)||(normdev<-
0.01));
	return res1;
}
void main(){
	float n;
	cin>>n;
	cout<<findroot(n)<<endl;
}
```

١- اكتب تابع complication بلغة c++ يقوم بحساب و تدوير عدد إلى عدد من مضاعفات ٢
٢- اكتب إجرائية print تقوم بطباعة تقريب جميع الأعداد إلى مضاعفات ٢ ضمن المجال المغلق [1..n] (مستفيدا من الطلب الأول)

```
int complication(int n){
	if(n%2==0)
		return n;
	else
		return n+(2-n%2);
}
void print(int n){
	for(int i=1;i<=n;i++)
		cout<<complication (i)<<endl;

}
void main(){
	print(10);

}
```

- المطلوب كتابة تابع pow1 لإيجاد X^n حيث x عدد كسري و n عدد صحيح موجب بشكل تكراري.

- المطلوب كتابة إجرائية pow2 لإيجاد X^n حيث x عدد حقيقي و n عدد صحيح موجب بشكل تكراري.

ملاحظة: **(ما الفرق بين التابعين pow1 و pow2)**

```
#include<iostream.h>
float pow1(float x,int n){
	float res=1;
	for(int i=1;i<=n;i++)
		res*=x;
	return res;
}
void pow2(float x,int n,float &res){
	res=1;
	int i=1;
	while(i<=n){
		res*=x;
		i++;
```

```
        }
}
void main(){
        int n;
        float x,res;
        cin>>x>>n;
        cout<<pow1(x,n)<<endl;
        pow2(x,n,res);
        cout<<res<<endl;
}
```

من أنواع الفرز التي مر معنا هو الفرز بالتعويم أو الفقاعات bubble sort و الآن نتطرف إلى نوع آخر من الفرز و هو الفرز بالاختيار selection sort و هو عبارة عن فكرة بسيطة و هي البحث عن أصغر عنصر في حال الترتيب التصاعدي ووضعه في بداية المصفوفة و نتابع عملية البحث في كل مرة عن العنصر الأصغر ثم الأصغر إلى أن نصل ترتيب عناصر المصفوفة .

عدد المقارنات في خوارزمية الاختيار هو:

N*(N-1)/2

أي إذا كان عدد عناصر الجدول(النسق)=7 هنا بحاجة إلى عدد من المقارنات هو 21 عملية مقارنة.

```
#include<iostream.h>
const int n=5;
struct array{
        int  a[n];
};
void sortselection(array&a,int size){
        for(int i=0;i<n-1;i++){
                int j=i;
                for(int k=i+1;k<n;k++)
                        if(a.a[k]<a.a[j])
                                j=k;
                        int temp=a.a[i];
                        a.a[i]=a.a[j];
                        a.a[j]=temp;
        }
}
void print(array a,int n){
        int i=0;
        while(i<n){
                cout<<a.a[i]<<",";
                i++;
        }
        cout<<endl;
}
void main(){
        array a={5,2,3,4,1};
         /*for(int i=0;i<n;i++)
                 a.a[i]=i+1;*/
         print(a,n);
         sortselection(a,n);
         print(a,n);

}
```

سبحانك ربنا لا علم لنا إلا ما علمتنا

إعداد المهندس خالد ياسين الشيخ
khaledyassinkh@gmail.com

اكتب تابع بلغة c++ يحول من درجة حرارة على سلم فاهرنهايت إلى القيمة المكافئة بالدرجة المئوية Celsius علماً أن قانون التحويل هو:

$$c=\frac{5}{9}+(f-32)$$

```
#include<iostream.h>
double c(double f){
        return (5.0/9)*(f-32);
}
void main(){
        double f;
        cin>>f;
        cout<<c(f)<<endl;
}
```

العودية (التراجعية) recursive:

العودية هي عملية استدعاء التابع لنفسه عدد **محدد** من المرات
العودية مكلفة زمنيا و ماديا لجهاز الحاسب حيث تعمل على استهلاك جزء كبير من ذاكرة الحاسب الرئيسية حيث كل استدعاء عودي يتعامل من قبل الذاكرة بفضاء مستقل عن الاستدعاءات العودية السابقة.
و نحن نعلم أن الحل التكراري يكون حلا أفضل من حيث الوقت و الذاكرة و لكننا نلجأ إلى الحل العودي في بعض الأحيان عندما يكون الحل التكراري غير واضح و من الناحية النظرية كل خوارزمية تكرارية يقابلها خوارزمية عودية و العكس صحيح.
ملاحظة: يكون الحل العودي مناسبا في الرسم العودي و سنتطرق للرسم العودي بالتفصيل الممل؟؟؟؟.
و المطلوب كتابة تابع بلغة c++ تابع يقوم بحساب العاملي لعدد مدخل

0!=1
1!=1
2!=2*1=2
5!=5*4*3*2*1=120
n!=n*(n-1)*(n-2)*..........2*1

```
#include<iostream.h>
int factorial(int x){
        if(x==0)
                return (1);
        else
                return (x*factorial(x-1));
}
void main(){
        int x;
        cin>>x;
        cout<<x<<"!="<<factorial(x)<<endl;
}
```

في المثال العودي يمكن أن نشبه ما يحصل بالذاكرة كالتالي:

output

First call to countdown Num is 2 — Countdown(1); — output: 2...

second call to countdown Num is 1 — Countdown(0); — output: 1...

third call to countdown Num is 0 — //no //recursive //call — output: blastoff!

```
#include<iostream.h>
void countdown(int num){
        if(num==0)
                cout<< "balstoff";
        else
        {
                cout<<num<<"...\n";
                countdown(num-1);
        }
}
void main(){
        countdown(2);
}
```

إعداد المهندس. خالد ياسين الشيخ

khaledyassinkh@gmail.com

المطلوب كتابة تابع عودي sum يقوم بجمع عددين صحيحين موجبين .
المطلوب كتابة تابع عودي mult يقوم بضرب عددين صحيحين موجبين.

```
#include<iostream.h>
int sum(int x,int y){
	if(y==0)
		return x;
	else
		return 1+sum(x,y-1);
}
int mult(unsigned int x,unsigned int y){
	if(y==0)
		return 0;
	else
		return x+mult(x,y-1);
}
void main(){
	int x,y;
	cin>>x>>y;
	cout<<sum(x,y)<<endl;
	cout<<mult(x,y)<<endl;
}
```

اكتب تابع sum عودي لحساب مجموع عناصر نسق صحيح.

```
#include<iostream.h>
#include<conio.h>
int sum(int a[],int n){
	if(n==0)
		return a[0];
	else
		return a[n]+sum(a,n-1);
}
void main(){
#define c 5
	int mat[c]={1,20,3,4,5};
	cout<<sum(mat,c-1);


}
```

- المطلوب كتابة تابع pow1 لإيجاد X^n حيث x عدد كسري و n عدد صحيح موجب بشكل عودي.

- المطلوب كتابة إجرائية pow2 عودية لإيجاد X^n حيث x عدد كسري و n عدد صحيح موجب بشكل.

ملاحظة: **(ما الفرق بين التابعين pow1 و pow2)**

```
#include<iostream.h>
double pow1(double x,int n){
	if(n>0)
		return x*pow1(x,n-1);
	else
		return 1;
}
void pow2(double x,int n,int &res){
	if(n>0){
		pow2(x,n-1,res);
		res=res*x;
	}
	else
		res=1;
```

<table>
<tr><td>

```
}
void main(){
	int x,n,res;
	cin>>x>>n;
	cout<<pow1(x,n)<<endl;
	pow2(x,n,res);
	cout<<res<<endl;
}
```

</td><td></td></tr>
<tr><td>

```
#include<iostream.h>
int f1(int n){
	if(n<0)
		return 10;
	else
		return n/2+f1(n-1);
}
void main(){
	int res=f1(f1(5));
	cout<<res<<endl;
}
```

</td><td>الخرج:
a. 16
b. 25
c. 60
d. 74
e. 62</td></tr>
<tr><td>

```
#include<iostream.h>
int f1(int n){
	if(n>0)
		return 0;
	else
		return n+f1(n+1);
}
int f2(int n){
	if(n<0)return f1(n);
	else
		return n*f1(-n);
}
void main(){
	int res=f1(4);
	cout<<res<<endl;
	cout<<f1(-3)<<endl;
	cout<<f2(5)<<endl;
	cout<<f2(-4)<<endl;
	cout<<f1(f2(2))<<endl;
}
```

</td><td>الخرج:
0
-6
-75
-10
-21</td></tr>
<tr><td>

```
#include<iostream.h>
int i=0;
void main(){
	char c[10]="khaled.";
	if(c[i]!='e')
	{
		i++;
		main();
		cout<<c[i]<<",";
```

</td><td>الخرج:
a. d,e,l,a,h,k,
b. e,l,a,h,k,
c. e,e,e,e,
d. e,l,a,h,k,
e. none of the above

علل السبب؟</td></tr>
</table>

khaledyassinkh@gmail.com
إعداد المهندس. خالد ياسين الشيخ

<table>
<tr><td>

```
	}
}
```

</td><td></td></tr>
<tr><td>

```
#include<iostream.h>
int gcd(int x, int y){
	if(x%y==0)
		return y;
	else
		return gcd(y,x%y);
}
int gcd1(int x,int y){
	return y==0?x:gcd1(y,x%y);
}
void main(){
	int x,y;
	cin>>x>>y;
	cout<<gcd(x,y);
}
```

</td><td>اكتب تابع بلغة البرمجة gcd يقوم بإيجاد القاسم المشترك الأعظم لعددين صحيحين موجبين بشكل عودي.
<u>لاحظ التابع gcd1 و كيفية كتابة التعليمة في سطر واحد.</u></td></tr>
<tr><td>

```
#include<iostream.h>
void swap(int a[],int k){
	int temp=a[k];
	a[k]=a[k+1];
	a[k+1]=temp;
}
void bubblesort(int *a,int n){
	for(int i=0;i<n-1;i++)
		for(int j=0;j<n-1;j++)
			if(a[j]>a[j+1])
				swap(a,j);
}
void print(int a[],int n,int i){
	if(i<n){
		print(a,n,i+1);
		cout<<a[i]<<",";
	}
}
void main(){
#define n 5
	int mat[n]={5,3,2,1,4};
	print(mat,n,0);
	cout<<endl;
	bubblesort(mat,n);
	print(mat,n,0);
	cout<<endl;
}
```

</td><td>اكتب بلغة c++ إجرائية عودية print بطباعة عناصر مصفوفة بشكل معكوس.
الخرج:
4,1,2,3,5,
5,4,3,2,1,</td></tr>
<tr><td>

```
#include<iostream.h>
int sum(int n,int a){
	if(n>0)
		return sum(n-1,a+n);
```

</td><td>خرج البرنامج هو:
a. 8
b. 12
c. 15</td></tr>
</table>

إعداد المهندس خالد ياسين الشيخ

khaledyassinkh@gmail.com

Code	Question
else return a; } void main(){ int x=sum(5,3); cout<<x<<endl; }	d. 17 **e. <u>18</u>**
#include<iostream.h> int sum(int x,int y){ if(x>0) return 1+sum(x-1,y); else return y; } void main(){ int x=sum(5,3); cout<<x<<endl; }	خرج البرنامج هو: a. 3 b. 7 c. 6 **d. <u>8</u>** e. 4
void strange(int n){ if(n>0){ strange(n/2); cout<<n%2; } } void main(){ strange(13); }	خرج البرنامج هو: a. 11011 b. 1011 **c. <u>1101</u>** d. 1111 e. 1010 عمل التابع هو: a. طباعة عدد صحيح بشكل ست عشري. b. طباعة عدد صحيح بشكل عشري. **c. <u>طباعة عدد صحيح بشكل ثنائي.</u>** d. طباعة عدد صحيح بشكل ثماني. e. طباعة أجزاء عدد صحيح .
int strange(int n,int i){ if(n==0) return 0; else return (n%10)*pow(2,i)+strange(n/10,i+1); } void main(){ cout<<strange(11111111,0); }	خرج البرنامج هو: a. 32 b. 64 c. 128 *d. <u>255</u>* e. 256
#include<iostream.h> int f1(int n,int y){ if(n==0||y==0) return n+y; else return n+y+f1(y-1,n-1); } void main(){	خرج البرنامج هو: a. 23 **b. <u>16</u>** c. 24 d. 12 e. 29

khaledyassinkh@gmail.com
إعداد المهندس خالد ياسين الشيخ

<table>
<tr><td>

```
int x=f1(4,3);
cout<<"x="<<x<<endl;
}
```

</td><td></td></tr>
<tr><td>

```
#include<iostream.h>
int area(int n){
        if (n==1)
        return 1;
        else
                return n+n-1+area(n-1);
}
void main(){
        int x=3;
        cout<<area(x)<<endl;
}
```

</td><td>
خرج البرنامج هو:
a. 3
b. 7
c. 8
d. 9
e. 10
f. 12
</td></tr>
<tr><td>

```
#include<iostream.h>
#include<math.h>
 f1(int n){
        int I;
        for(i=4;i<=n*n*n;i*=4)
                cout<<sqrt(i)<<",";
        cout<<endl;
}
void main(){
        f1(5);
}
```

</td><td>
الخرج:
a. 2,4,8,
b. 2,4,
c. 2,
d. 2,4,8,16,32,
e. جميع ما سبق غير صحيح
</td></tr>
<tr><td>

```
#include<iostream.h>
#include<math.h>
 f1(int n){
        int I, j=1;
        for(i=1;i<=n;i+=3);
        cout<<i<<endl;
}
void main(){
        f1(2010);
}
```

</td><td>
الخرج:
a. 2009
b. 2010
c. 2011
d. 2012
e. جميع ما سبق غير صحيح
ملاحظة: عدد مرات تنفيذ الحلقة هو 2010/3=670 و في كل مرة يتم زيادة العداد بمقدار 3 و منه 670*3=2010 و أول قيمة للعداد هي 1 ومنه قيمة العداد بعد الخروج من الحلقة هي 2010+1=2011
</td></tr>
<tr><td>

```
#include<iostream.h>
#include<math.h>
 f1(int n){
        int I, j=1;
        for(i=2;i<=n;i+=3);
        cout<<i<<endl;
}
void main(){
        f1(5);
}
```

</td><td>
الخرج:
a. 4
b. 8
c. 12
d. 14
e. 16
عدد مرات دخول الحلقة هي 5/3=2 ومنه قيمة العداد بعد الخروج من الحلقة هي 2*3+2=8
</td></tr>
<tr><td>

```
#include<iostream.h>
const int n=5;
 f1(int x){
         int i=x, a[n]={1,2,3,4,99},*p=(a+2);
```

</td><td>
الخرج:
a. 1,2,3,4,99,
b. 2,3,4,
c. 2,3,4,99,
</td></tr>
</table>

khaledyassinkh@gmail.com

إعداد المهندس خالد ياسين الشيخ

```
	while(p!=&a[5])
		cout<<*p++<<",";

 }
void main(){
	f1(5);
}
```

d. 2,3,4,
e. **<u>3,4,99,</u>**

الخرج:

a. **<u>5050</u>**
b. 4950
c. 4851
d. 3916
e. 2211

```
#include<iostream.h>
#include<math.h>
 inline int f1(int n){
	return n*(n+1)/2;
}
void main(){
	cout<<f1(100);
}
```

التابع f1 يقوم بحساب:

a. مجموع الأعداد من 1 إلى n-1.
b. مجموع الأعداد من 1 إلى n-2.
c. **<u>مجموع الأعداد من 1 إلى n.</u>**
d. مجموع الأعداد من n-2 إلى 1.
e. مجموع قواسم العدد n.

اكتب تابع عودي يقوم بقسمة عددين صحيحين موجبين و الناتج صحيح بطريقة الطرح المتكرر.

```
#include<iostream.h>
int mult(int x,int y){
	if(x==0|| x<y)
		return 0;
	else
		return 1+mult(x-y,y);
}
void main(){
	int x,y;
	cin>>x>>y;
	cout<<mult(x,y)<<endl;


}
```

اكتب تابع عودي لإيجاد أصغر عنصر في المصفوفة وحيدة الاتجاه.
اكتب إجرائية لإيجاد أصغر عنصر في نسق(شعاع)

```
int small(int *a,int size,int value,int index){
	if(size==1)
		return value;
	else

	if((a[index+1]<a[index]))
			value=a[index+1]
	return small(a,size-1,value,index+1);
}
```

إعداد المهندس خالد ياسين الشيخ

khaledyassinkh@gmail.com

<table>
<tr>
<td>اكتب تابع بلغة c++ يتحقق من كون عدد مدخل هو عدد أولي أم لا
ملاحظة: العدد الأولي لا يقبل القسمة إلا على نفسه و الواحد.</td>
<td>

```
#include<iostream.h>
bool isprime(int x){
	int i=2;
	while(x>i){
		if(x%i==0)
			return false;
			else
				if(x%i!=0)
					i++;
		}
		if (x==i)
			return true;
	}
	void main(){
		int x;
		cin>>x;
		if (isprime(x))
		cout<<x<<"is prime"<<endl;
		else
			cout<<x<<"is not
prime"<<endl;

	}
```

</td>
</tr>
<tr>
<td>اكتب إجرائية تقوم بطباعة جميع الأعداد الأولية ضمن المجال المغلق [1..x] حيث x عدد أكبر من 1 يتم إدخاله من قبل المستخدم.
ملاحظة: استفد من التابع isprime المكتوب أعلاه.</td>
<td>

```
#include<iostream.h>
bool isprime(int x){
	int i=2;
	while(x>i){
		if(x%i==0)
			return false;
			else
				if(x%i!=0)
					i++;
		}
		if (x==i)
			return true;
	}
void print(int n){
	for(int i=2;i<=n;i++)
		if(isprime(i))
	cout<<i<<" ";
		cout<<endl;
}

	void main(){
		int x;
		cin>>x;
		print(x);
	}
```

</td>
</tr>
</table>

khaledyassinkh@gmail.com

إعداد المهندس خالد ياسين الشيخ

اكتب إجرائية تقوم بطباعة جميع الأعداد الأولية ضمن المجال المغلق [1..x] حيث x عدد أكبر من 1 يتم إدخاله من قبل المستخدم.
ملاحظة: سوف نكتب البرنامج بأسلوب آخر.
الخرج:

2,3,5,7,11,

```
#include<iostream.h>
bool prime(int n){
	bool b=0;
	if(n==2)return 1;

	for(int i=2;i<n;i=i+1)
		if(n%i==0){
			b=0;
			break;}
		else
			b=1;
		return b;
}
void print(int n){
	int i=2;
	while(i<=n){
		if(prime(i))
			cout<<i<<",";
		i++;
	}
}
	void main(){
		int x=11;
		print(x);
	}
```

يعرف العدد الكامل بأنه العدد الذي يساوي مجموع قواسمه ماعدا نفسه
مثال: العدد 6 هو عدد كامل لأن مجموع قواسمه تساويه:

1+2+3=6

١- اكتب إجرائية divider لطباعة قواسم عدد.
١- اكتب بلغة البرمجة تابع check يتحقق من كون العدد كامل أم لا.
٢- إجرائية طباعة جميع الأعداد الكاملة الواقعة ضمن المجال المغلق [1..9000] مستفيدا من التابع check حصرا.
الخرج:

6
1,2,3,
28
1,2,4,7,14,
496
1,2,4,8,16,31,62,124,248,
8128
1,2,4,8,16,32,64,127,254,508,1016,2032,4064,

```
#include<iostream.h>
void divider(int n){
for(int i=1;i<=n/2;i++)
if(n%i==0)
cout<<i<<",";
cout<<endl;
}

bool check(int n){
	int sum=0;
	for(int i=1;i<=n/2;i++)
		if(n%i==0)
			sum+=i;
	if(n==sum)
		return true;
		else
		return false;
}
void complete(int max){

	for(int i=1;i<=max;i++){
		if(check(i)){
		cout<<i<<endl;
```

khaledyassinkh@gmail.com

إعداد المهندس. خالد ياسين الشيخ

```
                    divider(i);
                    }

          }
}
void main(){
          complete(9000);
}
```

نقول عن عددين أنهما صديقان n,m إذا كان مجموع قواسم العدد N هو M و مجموع قواسم العدد M هو N.

و العددان 220 و284 مثال على ذلك.

و المطلوب

١- اكتب بلغة البرمجة تابع check يتحقق من كون العددان صديقان أم لا.

٢- اكتب برنامج لإيجاد جميع الأعداد الصديقة ضمن المجال المغلق [1..4000] مستفيدا من التابع ckeck حصراً.

```
#include <iostream.h>
bool check(int n,int m){
          int s1=0,s2=0;
          for(int i=1;i<=n/2;i++)
          if(n%i==0)
          s1+=i;
          for(i=1;i<=m/2;i++)
          if (m%i==0)
                    s2+=i;
if(s1==m && s2==n)
          return 1;

          return 0;
}
void search(int max){
          int s1=0,s2=0;
          for(int n=1;n<=max;n++)
                    for(int i=1;i<=n;i++)
          if(check(n,i) &&n!=i)
                    cout<<n<<"  "<<i<<endl;

}
void main(){
          search(4000);
}
```

khaledyassinkh@gmail.com

إعداد المهندس. خالد ياسين الشيخ

اكتب تابع يتم فيه إدخال n عدد صحيح ثم يتم جمع هذه الأعداد المدخلة.

```
#include<iostream.h>
int addn(int n){
        int i=1,s=0;
        while(i++<=n){
        int x;
        cout<<"enter the number"<<i-1<<" ";
        cin>>x;
        s+=x;
        }
        return s;
}
void main(){
        int n;
        cin>>n;
        cout<<addn(n)<<endl;
}
```

الخرج:

```
*****
****
***
**
*
```

```
#include<iostream.h>
void print(int n){
        for(int i=5;i>=1;i--){
                for(int j=i;j>=1;j--)
                        cout<<'*';
                cout<<endl;
        }
}
void main(){
        print(5);
}
```

الخرج:

```
x=25
z=30.25
```

ملاحظة:

التعليمة template<typename t> تكافئ التعليمة template<class t>

و التعليمة typedef تستخدم لتعريف متحول على أنه نمط من معطيات معينة فمثلا typedef char c تدل على أن c هو نمط معطيات من نوع محرف. أي نستطيع الآن أن نصرح عن متحول على أنه من نمط c مثال c x; أي x متحول من نمط char

```
#include<iostream.h>
template <typename t>
t func(t n){
        return n*n;
}
void main(){
        int x;
         typedef double y;
        x=5;
        x=func(x);
        cout<<"x="<<x<<endl;
        y z;
        z=5.5;
        z=func(z);
        cout<<"z="<<z<<endl;
}
```

إعداد المهندس. خالد ياسين الشيخ
khaledyassinkh@gmail.com

ذكرنا في الجزء الأول من الكتاب مايلي:
struct عبارة عن بنية يمكن أن تحوي أنماط مختلفة من المعطيات (يمكن أن نشبه بالحقيبة التي يمكن تحوي أوراق و أقلام و دفاتر و حقيبة أخرى (حقيبة بقلب حقيبة) الخ) . و البرنامج يوضح ذلك.

```
#include<iostream.h>
struct date{
	int day;
	char month[30];
	int year;
};
struct person{
	char name[30];
	date Date;
};
struct mountain{
	char name[30];
	int height;
	person climber;
};
void main(){
mountain m={"Everset",
		8848,
{
	"Edmund hillary",
	{
		29,
			"may",
			1919
	}
	}
	};
if(m.climber.Date.year==1919)
cout<<"welcome to syria"<<endl;

}
```

الخرج:

Welcome ti syria

```
#include<iostream.h>
struct mountain{
	char name[30];
	int height;
} info={"everset", 8848};
void sayAbout(mountain);
void main(void){
	mountain k1;
	k1=info;
	sayAbout(k1);
}
void sayAbout(mountain m){
	cout<<m.name<<"
"<<m.height<<endl;
}
```

الخرج:

everset   8848

khaledyassinkh@gmail.com

إعداد المهندس. خالد ياسين الشيخ

#include<stdio.h> #include<string.h> struct tag{ char lname[20] ;/* last name*/ char fname[20];//first name float rate; }; struct tag my_struct; int main(void){ strcpy(my_struct.lname,"Omar"); strcpy(my_struct.fname,"Ali"); printf("%s ",my_struct.fname); printf("%s\n",my_struct.lname); return 0; }	الخرج: Ali Omar ملاحظة: يقوم التابع strcpy بأخذ و سيطين الوسيط الأول مصفوفة الحروف و التي تكون ذات طول مناسب لتخزين به قيمة الوسيط الثاني مع الحرف الصفري و الثاني يمثل سلسلة من الحروف يتم نسخها إلى الوسيط الأول مع الحرف الصفري . و هذا التابع موجود في المكتبة string.h لذلك يجب تضمينها قبل التعامل معه.
#include<stdio.h> #include<string.h> struct tag{ char lname[20] ;/* last name*/ char fname[20];//first name int age; float rate; }; struct tag my_struct; void show_name(struct tag* p); int main(void){ struct tag *st_ptr; st_ptr=&my_struct; strcpy(my_struct.lname,"Omar"); strcpy(my_struct.fname,"Ali"); printf("%s ",my_struct.fname); printf("%s\n ",my_struct.lname); my_struct.age=40; show_name(st_ptr); return 0; } void show_name(struct tag *p){ printf("%s ",p->fname); printf("%s ",p->lname); printf("%d\n",p->age); }	الخرج Ali Omar Ali Omar 40

إعداد المهندس خالد ياسين الشيخ
khaledyassinkh@gmail.com

```
#include<stdio.h>
int arr[10]={3,6,1,2,3,8,4,1,7,2};
void bubble(int *p,int n);
int compare(int *m,int *n);
void print(int a[],int n){
	for(int i=0;i<n;i++)
		printf("%d ",a[i]);
	putchar('\n');
}

void main(void){
	print(arr,10);
	bubble(arr,10);
	print(arr,10);
}
void bubble(int *p,int n){
	int i,j,t;
	for(i=n-1;i>=0;i--)
		for(j=1;j<=i;j++)
		if(compare(&p[j-1],&p[j]))
			{
				t=p[j-1];
				p[j-1]=p[j];
				p[j]=t;
			}
}
int compare(int *m,int *n){
	return (*m>*n);
}
```

الخرج:

3 6 1 2 3 8 4 1 7 2
1 1 2 2 3 3 4 6 7 8

```
#include<stdio.h>
int arr[10]={3,6,1,2,3,8,4,1,7,2};
void bubble(int *p,int n);
int compare(void *m,void *n);
void print(int a[],int n){
	for(int i=0;i<n;i++)
	printf("%d ",a[i]);
	putchar('\n');
}

void main(void){
	print(arr,10);
	bubble(arr,10);
	print(arr,10);
}
void bubble(int *p,int n){
	int i,j,t;
	for(i=n-1;i>=0;i--)
		for(j=1;j<=i;j++)
		if(compare(&p[j-1],&p[j]))
```

الخرج:

3 6 1 2 3 8 4 1 7 2
1 1 2 2 3 3 4 6 7 8

قارن التابع compare مع التابع compare في البرنامج أعلاه.

اختر الإجابة الصحيحة:

الفرق بين الفرز الداخلي و الترتيب(الفرز) الخارجي هو:

a. لا يوجد فرق بينهما
b. الترتيب الخارجي هو الترتيب الذي يحدث خارج الذاكرة الرئيسية
c. الترتيب الداخلي هو ما يحدث في الذاكرة الرئيسية عندما تكون حجم البيانات ليست كبيرة بحيث يمكن استيعابها بالذاكرة الرئيسية
d. **(*)b. و c.**
e. كل ما سبق خاطئ

<table>
<tr><td><pre>
                {
                        t=p[j-1];
                        p[j-1]=p[j];
                        p[j]=t;
                }
}
int compare(void *m,void *n){
        int *m1,*n1;
        m1=(int *)m;
        n1=(int *)n;
        return (*m1>*n1);
}
</pre></td><td></td></tr>
<tr><td><pre>
#include<iostream.h>
int main(){
int x=7,y=6;
void *p1=&x,*p2=&y;
int *m1,*m2;
p1=&y; p2=&x;
m1=(int *) p1;
m2=(int *)p2;
*m1=*m2+3;
*m2=*m1+5;
cout<<*m1<<endl;
cout<<*m2<<endl;

return 0;
}
</pre></td><td>الخرج:
10
15</td></tr>
<tr><td><pre>
#include<stdio.h>
void main(void){

        char a[5];
        printf("   array=%p\n&array[0]=%p\n
&array=%p\n",a,&a[0],&a);
}
</pre></td><td>الخرج:
array=0012FF78
&array[0]=0012FF78
&array=0012FF78</td></tr>
</table>

khaledyassinkh@gmail.com إعداد المهندس خالد ياسين الشيخ

<table>
<tr>
<td>

```
#include<stdio.h>
#include<iostream.h>
void main(void){

int x;
cout<<&x<<endl;
        char a[5];
        printf("%p \n",&x);
}
```

</td>
<td>

الخرج:

0x0012FF7C
0012FF7C

</td>
</tr>
<tr>
<td>

```
#include<iostream.h>
void print(int x,int y){
        do{
                cout<<"your number
"<<x<<endl;
                x++;
        }
        while(x<=y);
}
void main(){
        int a,b;
        cin>>a>>b;
        if(a>b){
a*=b;
        b=a/b;
        a=a/b;}

        print(a,b);
}
```

</td>
<td>

اكتب برنامج بلغة c++ تقوم بطباعة الأعداد الصحيحة المحصورة بين x و y أي ضمن المجال المغلق [x..y]
مثال: x=3 و y=5 أو x=5 و y=3 (اجعل محتوى x تحمل الرقم الأصغر دوماً) يتم طباعة مايلي:

your number 3
your number 4
your number 5

</td>
</tr>
<tr>
<td>

```
#include<iostream.h>
void watch(float t){
        int x=1;
        while(t<=40){
                if(t==39){
                        for(;;){
cout<<"warning!!!!!!!!!!\a\a\a"<<endl;

        cout<<"temperature ="<<t<<endl;

                if(x%15==0){
                cout<<x<<endl;
                break;   }
                cout<<x<<endl;
                ++x;
                }}
```

</td>
<td>

يوجد في مشتشفى المواساة عدد من المرضى في غرفة واحدة و في هذه الغرفة يوجد طبيب واحد فقط و من الصعب على هذا الطبيب مراقبة هؤلاء المرضى جميعا في آن واحد.
أكتب برنامج لتحذير الطبيب بوجود خطر على حياة إحدى المرضى و ذلك بسبب ارتفاع درجة حرارته إلى 39 و هذه الدرجة تعتبر خطرة على حياة المريض مما يستدعي مساعدة عاجلة من الطبيب لتخفيض درجة حرارة المريض
بحيث يكون هذا التحذير على شكل صوت تنبيه يخرج من جهاز الكمبيوتر.
مع العلم أنه يتم قياس درجة حرارة المريض بفارق نصف درجة في كل مرحلة.
نحن نعلم أن درجة الإنسان الطبيعية هي ٣٧
عدل البرنامج بحيث يتم إصدار صوت لمدة خمس دقائق و في حال إدخال أي محرف من لوحة المفاتيح من قبل الطبيب يتوقف إصدار الصوت.

</td>
</tr>
</table>

khaledyassinkh@gmail.com
إعداد المهندس خالد ياسين الشيخ

```
        t=t+0.5;
    }
}
    void main(){
        watch(36);
    }
```

```
#include<iostream.h>
void main(){
    int mat[3][3]={1,2,3,4,5,6,7,8,9};
    for(int i=0;i<3;i++){
        for(int j=0;j<3;j++)
            if(i>j)

    cout<<mat[i][j]<<" ";
}
}
```

ما هو عمل البرنامج:

a. **طباعة عناصر ما تحت القطر الرئيسي**
b. طباعة عناصر ما فوق القطر الرئيسي
c. طباعة عناصر ما تحت القطر الثانوي
d. طباعة عناصر ما فوق القطر الثانوي
e. a. و b.

```
#include<iostream.h>
void main(){
    int mat[3][3]={1,2,3,4,5,6,7,8,9};
    for(int i=0;i<3;i++){
        for(int j=0;j<3;j++)
            if(i<j)

    cout<<mat[i][j]<<" ";
}
}
```

ما هو عمل البرنامج:

a. طباعة عناصر ما تحت القطر الرئيسي
b. **طباعة عناصر ما فوق القطر الرئيسي**
c. طباعة عناصر ما تحت القطر الثانوي
d. طباعة عناصر ما فوق القطر الثانوي
e. a. و b.

إعداد المهندس خالد ياسين الشيخ
khaledyassinkh@gmail.com

```
#include<stdio.h>
void main(){
        int x;//input  9
        printf("enter the value x:");
        scanf("%d",&x);
        printf("number is=%d\n",x);
}
```

الخرج:

enter the value x:9
number is=9

ملاحظة: تعليمة scanf تعليمة لإدخال المعلومات من لوحة المفاتيح keyboard

```
#include<stdio.h>
void main(){
        int x,y; //input x=9 , y=10
        printf("enter the value x and y:");
        scanf("%d%d",&x,&y);
        printf("%d + %d= %d\n",x, y, x+y);
}
```

الخرج:

enter the value x and y:9 10
9 + 10=19

```
#include<iostream.h>
void main(){
        int x=7,y=1;
        cout<<(y<<x)<<endl;;
        cout<<(x>>y)<<endl;
        y=(y<<x);
        x=(x>>y);
        cout<<y<<endl;
        cout<<x<<endl;

}
```

الخرج:

128
3
128
3

ملاحظة: المؤثر << يسمى بمؤثر الإزاحة اليساري left shift
و المؤثر >> يسمى بمؤثر الإزاحة إلى اليمين Right shift .
و التعليمة y=(y<<x) تعني إزاحة المتحول y بمقدار 7 بتات(خانات).

0	0	0	0	0	0	0	1

تصبح قيمة y=128 بعد الإزاحة إلى اليسار بسبع خانات كالتالي:

1	0	0	0	0	0	0	0

و التعليمة (x>>y)=xتعني إزاحة المتحول x=7 بمقدار 1 بت إلى اليمين:

0	0	0	0	0	1	1	1

تصبح بعد الإزاحة قيمة x=3 كالتالي:

0	0	0	0	0	0	1	1

```
#include<iostream.h>
void main(){
        int a=5,b=6;
        cout<<(a>b)<<endl;
        cout<<!(a>b)<<endl;
        cout<<(a&&b==a)<<endl;
        cout<<(a && b)<<endl;
```

الخرج:

0
1
0
1
4
3
1

khaledyassinkh@gmail.com

إعداد المهندس. خالد ياسين الشيخ

<table>
<tr><td>

```
        cout<<(a&b)<<endl;
        cout<<(a^b)<<endl;
        cout<<(a||b)<<endl;
        cout<<(a|b)<<endl;
        cout<<(a>>1)<<endl;
        cout<<(a<<2)<<endl;
        cout<<!a<<endl;
        cout<<~a<<endl;
b=-7;
        cout<<~b<<endl;

}
```

</td><td>7
2
20
0
-6
6</td></tr>
<tr><td>

```
#include<iostream.h>
#include<stdio.h>
#include<time.h>
void delay(int sec);
void main(){
        int wait;
        cout<<"enter wait";
        cin>>wait;
        delay(wait);
        cout<<"time........."<<endl;
}
void delay(int sec){
        int s;
        for(;;){
                s=clock()/CLOCKS_PER_SEC;
                if(s==sec)
                        break;
        }
}
```

</td><td>الدالة delay تقوم بالانتظار لوقت محدد بالثواني</td></tr>
<tr><td>

```
#include<iostream.h>
void sort(int b[],int size){
        for(int i=0;i<size-1;i++)
                if(b[i]>b[i+1]){
                        int temp=b[i];
                        b[i]=b[i+1];
                        b[i+1]=temp;
                }
                if(size>1)
                        sort(b,size-1);

}
void readarray(int b[],int i,int size){
        if(i<size){
                cout<<"enter the
b["<<i<<"]"<<endl;
                cin>>b[i];
```

</td><td>اكتب إجراء sortعودي لخوارزمية الفرز بالفقاعات.
اكتب إجراء print عودي لطباعة عناصر نسق.
اكتب إجراء readarray لقراءة عناصر نسق بشكل عودي.
ملاحظة: يجب على المستخدم للبرنامج تحديد حجم النسق.</td></tr>
</table>

khaledyassinkh@gmail.com

إعداد المهندس خالد ياسين الشيخ

```
            readarray(b,i+1,size);
        }
}
void print(int b[],int i,int size){
        if(i<size){
            cout<<b[i]<<",";
            print(b,i+1,size);
        }
}
void main(){
        cout<<"enter number size
array"<<endl;
        int c;
        cin>>c;
#define n c
        int * b=new int[n];
        readarray(b,0,n);
        cout<<endl;
        print(b,0,n);
        sort(b,n);
        cout<<"\nafter sort"<<endl;
        print(b,0,n);
delete[]b;
        cout<<endl;
}
```

```
#include<iostream.h>
int f1(int **p,int c){
        int res=0;
        for(int i=0;i<c;i++)
            for(int j=0;j<i;j++)
                res=res+*(*(p+i)+j);

        return res;
}
void print(int **p,int n){
        int i,j;
        for(i=0;i<n;i++){
            for( j=0;j<n;j++)

        cout<<*(*(p+i)+j)<<"\t";
                cout<<endl;
}
}


void main(){
        int c=3;
        int **a;
        a=new int *[c];
```

الخرج:

```
0	1	2
1	2	3
2	3	4
f1=6
```

درجة تعقيد التابع f1 هو:

a. **<u>$o(n^2)$</u>**
b. $o(n)$
c. $o(n/2)$
d. $o(n^4)$
e. $\log_2(n^2)$

عدد عمليات الجمع هي:

a. n^2+n
b. n^2+n
c. $\frac{n2+n}{2}$
d. $\frac{n2-n}{2}$
e. $\frac{n2+n}{4}$

عمل التابع f1 هو:

a. جمع عناصر مصفوفة المربعة.
b. **<u>جمع عناصر ما تحت القطر الرئيسي في المصفوفة.</u>**
c. جمع عناصر ما فوق القطر الثانوي في المصفوفة.
d. جمع عناصر محيط المصفوفة.
e. الجواب الصحيح يختلف عما سبق.

	` for(int i=0;i<c;i++)` `  a[i]=new int[c];` ` for(i=0;i<c;i++)` `  for(int j=0;j<c;j++)` `   *(*(a+i)+j)=i+j;` `  print(a,c);` `  cout<<"f1="<<f1(a,c)<<endl;` `  for(i=0;i<c;i++)` `   delete[]a[i];` `  delete[]a;` `}`
الخرج: 36 التابع p1 يقوم بحساب x^n حيث x عدد كسري و n عدد صحيح موجب.	`#include<iostream.h>` `double p1(double x,unsigned int n){` ` double res=1;` ` while(n>0){` `  if(n%2!=0) res=res*x;` `  x=x*x;` `  n=n/2;` ` }` ` return res;` `}` `void main(){` ` double x=6;` ` int n=2;` ` cout<<p1(x,n)<<endl;` `}`
اكتب برنامج بلغة c++ للتعامل مع الأعداد العقدية و المطلوب: ١- إجرائية read لقراءة عدد عقدي. ٢- إجرائية(دالة) print لطباعة عدد عقدي. ٣- تابع(دالة) sum لحساب مجموع عددين عقديين. ٤- تابع sub لحساب طرح عددين عقديين. ٥- تابع mult لحساب ضرب عددين عقديين. ٦- تابع div لحساب قسمة عددين عقديين. ٧- تابع t لحساب طويلة عدد عقدي.	`#include<iostream.h>` `#include<math.h>` `struct num{` ` float re;` ` float im;` `};` `void read(num&);` `void print(num&);` `num sum(num&,num&);` `num sub(num&,num&);` `num mult(num&,num&);` `num div(num&,num&);` `double t(num);` `void main(){` ` num x,y,z;` ` read(x);` ` read(y);` ` cout<<"----------\n";` ` print(x);` ` cout<<endl;` ` print(y);`

```
        cout<<"-----------\n";

        z=sum(x,y);
        cout<<"sum=";
        print(z);
                cout<<endl;
        z=sub(x,y);
        cout<<"sub=";print(z);cout<<endl;
        z=mult(x,y);
        cout<<"mult=";print(z);cout<<endl;
        z=div(x,y);
        cout<<"div=";
        print(z);cout<<endl;
        cout<<t(x)<<endl;
        cout<<t(y)<<endl;

}

void read(num&x){
        cin>>x.re;
        cin>>x.im;


}
void print(num& x){
        cout<<x.re<<"+"<<x.im<<"I";}

num sum(num&x,num&y){
        num x1;
        x1.re=x.re+y.re;
        x1.im=x.im+y.im;
        return x1;
}
num sub(num&x,num&y){
        num x1;
        x1.re=x.re-y.re;
        x1.im=x.im-y.im;
        return x1;
}
num mult(num&x,num & y){
        num x1;
        x1.re=(x.re*y.re)-(x.im*y.im);
        x1.im=(x.re*y.im)+(y.re*x.im);
        return x1;
}
num div(num&x,num&y){
        num x1,y1,z;
        float m;
        y1.re=y.re;
```

إعداد المهندس. خالد ياسين الشيخ
khaledyassinkh@gmail.com

```
	y1.im=-y.im;
	x1=mult(x,y1);
	m=pow(y.re,2)+pow(y.im,2);
	z.re=x1.re/m;
	z.im=x1.im/m;
	return z;
}
double t(num x){
	return sqrt(pow(x.re,2)+pow(x.im,2));

}
```

اكتب بلغة ++c برنامج للتعامل مع التاريخ على اعتبار أن التاريخ يمثل بـــــ d اليوم و m الشهر و y السنة.

١- تابع read لقراءة تاريخ.
٢- إجرائية printdate لطباعة تاريخ بالشكل d/m/y.
٣- تابع compare2date يقوم بالمقارنة بين تاريخين و يعيد القيمة 1 في حال كان التاريخ الأول أكبر تماما من التاريخ الثاني و إلا فإنه يعيد القيمة 0.

ملاحظة:
الإدخال يتم على الشكل التالي:
إدخال اليوم d
إدخال الشهر m
إدخال السنة y

```
#include<iostream.h>
struct date{
	int d,m,y;
};
date read(date);
void printdate(date);
bool compare2date(date,date);
int main(){
	date x,y;
	x=read(x);
	y=read(y);
	printdate(x);
	printdate(y);
	cout<<"--------\n";
	cout<<compare2date(x,y)<<endl;
}

date read(date x){
	cin>>x.d;
	cin>>x.m;
	cin>>x.y;
	return x;
}
void printdate(date x){
	cout<<x.d<<"/"<<x.m<<"/"<<x.y<<end
l;
}
bool compare2date(date x,date y){
	if(x.y>y.y)
	return true;
	else
		if(x.m>y.m)
			return true;
			else
			  if(x.d>y.d)
```

إعداد المهندس خالد ياسين الشيخ
khaledyassinkh@gmail.com

```
                    return 1;
                        else
                    return false;
}
```

اكتب برنامج بلغة ++c يقوم بتخزين بيانات طلاب السنة الأولى في كلية الهندسة المعلوماتية بجامعة دمشق حيث يعرف كل طالب من خلال رقم ذاتي number(لا يتكرر) و اسم name بالإضافة إلى علامات المواد التالية:

الرياضيات math
البرمجة١ prog1
الإنكليزي en
العربي ar

و العلامة النهائية من 100 و جميع العلامات صحيحة.

<u>ملاحظة يجب معالجة حالة تكرار إدخال رقم طالب موجود .</u>

والمطلوب:

١- اقترح بنية معطيات مناسبة لتخزين بيانات الطالب.
٢- إجرائية read لقراءة البيانات الخاصة بكل طالب.
٣- إجرائية sortstudent تقوم بترتيب عناصر بيانات الطلاب تنازلياً حسب المعدل.
٤- إجرائية printstudent لطباعة اسم الطالب و رقمه و معدله.

```
#include<iostream.h>
struct student{
        int number;
        char name[30];
        int prog1;
        94n tar;
        int en;
        int math;
        float av;
};
void read(student a[],int n){
        for(int i=0;i<n;i++){
                cin>>a[i].number;
                for(int j=0;j<=i-1;j++){

        if(a[j].number==a[i].number){

        cout<<"replay"<<endl;

        cin>>a[i].number;

        while(a[j].number==a[i].number)
                                {

        cout<<"replay";

        cin>>a[i].number;
                                }
                        }
                }
                cin>>a[i].name;
                cin>>a[i].math;
                cin>>a[i].prog1;
                cin>>a[i].en;
                cin>>a[i].ar;
```

```
        a[i].av=(a[i].math+a[i].prog1+a[i].en+a
[i].ar)/4.0;
}
}
void sortstudent(student a[],int n){
        for(int i=1;i<n;i++)
                for(int j=0;j<n-1;j++)
                        if(a[j].av<a[j+1].av)
                        {
                                student
temp=a[j];
                                a[j]=a[j+1];
                                a[j+1]=temp;
                        }
}
void printstudent(student *a,int n){
        for(int i=0;i<n;i++)
        cout<<a[i].number<<"\t"<<a[i].name<
<"\t"<<a[i].av<<endl;
}

void main(){
#define max 3
        student a[max];
        read(a,max);
        sortstudent(a,max);
        printstudent(a,max);
}
```

لدينا بيانات تتضمن معلومات عن موظفي كلية الهندسة المعلوماتية بجامعة دمشق تتضمن بيانات الموظف المعلومات التالية:
رقم ذاتي number علما أن رقم الموظف يعطى بشكل تسلسلي حصراً(أي الموظف الذي يتم إدخال بياناته أولا يأخذ الرقم 1 و الموظف الثاني الرقم 2 و هكذا....)
الاسم الأول first_name
الكنية last_name
تاريخ الميلاد و يتضمن اليوم d و الشهر m و السنة y
الراتب salary
و المطلوب كتابة برنامج بلغة البرمجة c++ :

١- اقترح بيئة معطيات مناسبة لتخزين بيانات الموظف.
٢- تابع لقراءة بيانات الموظف.
٣- إجرائية تقوم بترتيب بيانات الموظفين ترتيباً تنازلياً حسب الراتب.
٤- إجرائية لطباعة بيانات الموظفين على شاشة الحاسب.
٥- عدل الطلب الثاني بحيث تصبح التابع إجرائية أي تابع لا يعيد قيمة

```
#include<iostream.h>
 struct birthday{
                int d,m,y;

        };
struct employee{
        int number;
        char first_name[30];
        char last_name[30];
        birthday birth;
        double salary;
};
employee  read(employee e){
        static int i=1;
        e.number=i++;
        cout<<"number=";cout<<e.number<<
endl;
        cout<<"first_name=";
cin>>e.first_name;
        cout<<"last_name=";
```

الطلب الخامس:

```
void  read(employee & e){
        static int i=1;
        e.number=i++;
```

إعداد المهندس خالد ياسين الشيخ
khaledyassinkh@gmail.com

```
        cout<<"number=";cout<<e.number<<endl;
        cout<<"first_name="; cin>>e.first_name;
        cout<<"last_name="; cin>>e.last_name;
cout<<"birthday"<<"************"<<endl<<"\tday=";
        cin>>e.birth.d;
        cout<<endl<<"\tmonth="; cin>>e.birth.m;
        cout<<endl<<"\tyear="; cin>>e.birth.y;
        cout<<endl<<"salary=";cin>>e.salary;

}
```

في البرنامج(التابع) الرئيسي نكتب الاستدعاء على الشكل:

```
        read(e[i]);
```

ملاحظة:سأشرح بالتفصيل الملفات و كيفية التعامل معها بلغة C++ في قادم الصفحات أو (في الجزء الثالث).

```
cin>>e.last_name;
        cout<<"birthday"<<"************"
<<endl<<"\tday=";
        cin>>e.birth.d;
        cout<<endl<<"\tmonth=";
cin>>e.birth.m;
        cout<<endl<<"\tyear=";
cin>>e.birth.y;
        cout<<endl<<"salary=";cin>>e.salary;
        return e;
}
void sortemp(employee e[],int n){
        for(int i=0;i<n-1;i++)
                for(int j=0;j<n-1;j++)

        if(e[j].salary<e[j+1].salary){
                                employee
temp=e[j];
                        e[j]=e[j+1];
                        e[j+1]=temp;
}
}
void printemp(employee e[],int n){
        for(int i=0;i<n;i++){
                cout<<"number=";
                cout<<e[i].number<<endl;
                cout<<"firsr_name=";
                cout<<e[i].first_name<<endl;
                cout<<"last_name=";
                cout<<e[i].last_name<<endl;
                cout<<"birthday=";
                cout<<e[i].birth.d<<"-
"<<e[i].birth.m<<"-"<<e[i].birth.y<<endl;

        cout<<"salary="<<e[i].salary<<endl<<"
-----------"<<endl;

        }
}
void main(){
        const int max=3;
        employee e[max];;
        for(int i=0;i<max;i++)
                e[i]=read(e[i]);
        sortemp(e,max);
        printemp(e,max);


}
```

إعداد المهندس. خالد ياسين الشيخ
khaledyassinkh@gmail.com

```
#include<iostream.h>
void main(){
        int x=5;
        int *p=&x;
        cout<<p<<endl;
        cout<<&x<<endl<<&p<<endl;
        p=NULL;
        cout<<p<<endl;
        cout<<&p<<endl;
//*p=10 error خطأ

}
```

ليكن البرنامج المكتوب على يسارنا بلغة c++ و ليكن ما يلي:

اسم المتحول	memory address
x	0x0012FF7C
p	0x0012FF78

و المطلوب خرج البرنامج.

الخرج هو:

```
5
5
0x0012FF7C
0x0012FF7C
0x0012FF78
0x00000000
0x0012FF78
```

اختر الإجابة الأكثر **دقة**:

a. المؤشرات عبارة عن متحولات تحوي قيم في الذاكرة.
b. **(*)المؤشرات عبارة عن متحولات تحوى عناوين في الذاكرة.**
c. المؤشرات عبارة عن متحولات تؤشر إلى عناوين بالذاكرة.
d. المؤشرات عبارة عن متحولات تقوم بعملية تأشير إلى محتوى موقع بالذاكرة.
e. المؤشرات عبارة عن متحولات تحوي مؤشرات في الذاكرة.

```
#include<iostream.h>
void main(){
        int *p1=new int,*p2=new int;
        *p1=5;
        p2=p1;
        cout<<p1<<endl;
        cout<<p2<<endl;
        cout<<*p1<<endl;
        cout<<*p2<<endl;

        delete(p2);
        if(p2!=NULL){
        cout<<p1<<endl;
        cout<<p2<<endl;
        *p1=99;
        cout<<*p1<<endl;
        cout<<*p2<<endl;
        cout<<&p1<<endl;
        cout<<&p2<<endl;}

}
```

بفرض لدينا :

اسم المتحول	Memory address
P1	0x0012FF7C
P2	0x0012FF78

علما أن p1 يؤشر إلى العنوان 0x00421FF0 في الذاكرة و المطلوب خرج البرنامج.

الخرج:

```
0x00421FF0
0x00421FF0
5
5
0x00421FF0
0x00421FF0
99
99
0x0012FF7C
0x0012FF78
```

اختر الإجابة الصحيحة:

لنفرض أنه لدينا int *p=new int(5) فإنه عند تطبيق التعليمة delete(p) فهذا يعني:

a. المؤشر p أصبح لا يؤشر إلى أي عنوان في الذاكرة.
b. المؤشر p أصبح يؤشر إلى العنوان صفر في الذاكرة.
c. المؤشر p بقي يؤشر إلى نفس العنوان في الذاكرة.
d. يتم حذف محتوى *p من الذاكرة.
e. **(*)c. و d.**

khaledyassinkh@gmail.com

إعداد المهندس خالد ياسين الشيخ

<table>
<tr><td>

```
#include<iostream.h>
void main(){
        int *p1=new int,*p2=new int;
        *p1=5;
        p2=p1;
        delete(p2);

        cout<<*p1<<endl;

}
```

</td><td>

خرج البرنامج هو:

a. 5
b. 10
c. 20
d. <u>None of the above</u>

</td></tr>
<tr><td>

```
#include<iostream.h>
struct m{
                int y;
        };
struct k{
        int x,y;
        m t;

};
k *f(k *r){
        r->t.y=99;
        r->x=55;
        r->y=(*r).x+(*r).t.y;
return r;
}
void main(){
        k * r;
        k x;
        r=&x;
        f(&x);
        cout<<r->t.y<<",";
        cout<<r->x<<",";
        cout<<r->y<<endl;

}
```

</td><td>

خرج البرنامج هو:

a. <u>99,55,154</u>
b. 55,99,154
c. 154,55,99
d. 55,99,154
e. None of the above

</td></tr>
</table>

إعداد المهندس. خالد ياسين الشيخ

khaledyassinkh@gmail.com

اكتب برنامج بلغة c++ يقوم بتعريف أنماط المعطيات و التوابع و الإجرائيات التي تسمح بقراءة مجموعة من الأسماء مع الأرقام الذاتية الخاصة بها و تخزينها في سلسلة ديناميكية المطلوب:

١- إجرائية readinsert تقوم ببناء السلسلة مرتبة تصاعديا حسب رقم الطالب(علماً أن رقم الطالب لا يتكرر و أرقام هي أرقام صحيحة موجبة أكبر تماما من الصفر حصرا).
٢- إجرائية del تقوم بحذف العنصر الموافق لرقم ذاتي هو دخل الإجرائية مع معالجة محاولة حذف رقم طالب غير موجود.
٣- إجرائية addnew تقوم بإضافة عنصر جديد (اسم ورقم الطالب)إلى السلسلة مع معالجة عدم إضافة رقم طالب موجود.
٤- إجرائية printlist تقوم بطباعة عناصر السلسلة على الشكل التالي studentnmae studentnumber.
٥- تابع counterlist يقوم بإعادة عدد عناصر السلسلة.
٦- إجرائية movetofile تقوم بنقل عناصر السلسلة مرتبة تصاعديا حسب الرقم الطالب إلى ملف تسلسلي f.dat مع مراعاة حذف العنصر المنقول مباشرة.

ملاحظة:التعليمة p->number تكافئ التعليمة (*p).number و يجب تضمين p* بين قوسين لأن النقطة . لها أولوية أعلى من *

```
#include<iostream>
#include<string>
#include<fstream>
using namespace std;
typedef struct student * pstudent;
struct student{
        string name;
        int number;
        pstudent next;
};
bool search(pstudent p,int k){
        while(p!=0){
                if(p->number==k)
                        return 1;
                p=p->next;
        }
        return 0;
}


void readinsert(pstudent& p,student e){
        if(p==NULL /*==0*/){
                p=new(student);
        e.next=NULL;
                *p=e;}
        else
                if (p->number>e.number){/*((*p).number>e.number)*/
                        e.next=p;
                        p=new(student);
                        *p=e;
                }
                else
                {
                        pstudent prev,temp=p;

        while((temp!=0)&&(temp->number<e.number)){
                                prev=temp;
                                temp=temp->next;
                        }
                        e.next=temp;
                        prev->next=new (student);
                        *prev->next=e;
                }
```

khaledyassinkh@gmail.com

إعداد المهندس. خالد ياسين الشيخ

```
}
void del(pstudent&p,student e){
        if(p!=0)
                if(p->number==e.number){
                        pstudent pr=p;
                        p=p->next;
                        delete(pr);
                }
                else
                {
                        pstudent
temp=p,prev;

        while((temp!=0)&&(temp-
>number<e.number)){
                                prev=temp;
                                temp=temp-
>next;

                        }
                        prev->next=temp-
>next;
                        delete(temp);
                }
}

void addnew(pstudent &p,student e){
        if(p->number>e.number){
                e.next=p;
                p=new student;
                *p=e;
        }
        else
        {
                pstudent temp=p,prev;
                while((temp!=NULL)&&(temp-
>number<e.number)){
                        prev=temp;
                        temp=temp->next;
                }
                e.next=temp;
                prev->next=new(student);
                *prev->next=e;
        }


}
void printlist(pstudent p){
        cout<<"numberstudent"<<"\t"<<"nma
```

```
estudent"<<endl;
	while(p!=0){
		cout<<p->number<<"\t"<<p-
>name<<endl;
		p=p->next;
	}
}


int counterlist(pstudent p){
	if(p==0)
		return 0;
	else
		return 1+counterlist(p->next);
}

void movetofile(pstudent& p){
   pstudent temp=p;
	ofstream outfile("f.dat",ios::out);
	while(p!=NULL){
		temp=p;
		outfile <<p->number<<" "<<p-
>name<<endl;
		p=p->next;
		delete(temp);
	}
}
void main(){
#define n 3
	pstudent p=0; student e;
	for(int i=1;i<=n;i++){
		cin>>e.number;
		while(e.number<=0){
			cout<<"enter the
number student>=1"<<endl;
			cin>>e.number;
		}
		cin>>e.name;
		readinsert(p,e);
	}
	printlist(p);
	cout<<"number of student
is="<<counterlist(p)<<endl;
	cout<<"d"<<endl;
	cin>>e.number;
	if(search(p,e.number))
	del(p,e);
	else
		cout<<"not found";
```

إعداد المهندس خالد ياسين الشيخ
khaledyassinkh@gmail.com

```
	printlist(p);
	cout<<"add"<<endl;
	cin>>e.number;
	while(search(p,e.number))
	{
		cout<<"please enter the 
number"<<endl;
		cin>>e.number;
	}
	while(e.number<=0){
			cout<<"enter the 
number student>=1"<<endl;
			cin>>e.number;
		}

	cin>>e.name;

	addnew(p,e);
	printlist(p);
	movetofile(p);
}
```

اكتب برنامج بلغة c++ يقوم بالتعامل مع السلاسل (لقوائم) الخطية وحيدة الاتجاه حيث نفرض انه لدينا البنية التالية:

```
struct info{
int num;
info * next;
};
```

و المطلوب:

١- تابع insert لإضافة أو لحشر عنصر ضمن السلسلة.
٢- تابع del لحذف عنصر من السلسلة.
٣- تابع search للبحث عن عنصر ضمن السلسلة.
٤- إجرائية print لطباعة عناصر السلسلة.

كل عنصر من السلسلة له الحقول التالية:

num	next

حيث الحقل next هو مؤشر إلى العنصر التالي في السلسلة أن وجد و في حال أنه لا يؤشر إلى العنصر تالي فإن قيمته تساوي nil أي:

num	☒

و يمكننا تخيل شكل السلسلة في الذاكرة كما يلي:

بداية السلسلة — نهاية السلسلة

num next → num next ······→ num next

```
#include<iostream.h>
#include<conio.h>
struct info{
	int num;
	info * next;
};
info * insert(info*,info);
info* del(info*,info);
bool search(info*,int);
void print(info*);
char choice();

void main(){
	info* head=NULL;
	info v;
	char c;
	do{
		c=(choice());
		switch(c){
		case 'I':
		case 'i':{
			cout<<"enter the 
number";cin>>v.num;

	head=insert(head,v);break;}
		case 'D':
		case'd':{cout<<"enter the 
number";cin>>v.num;
```

```
                    head=del(head,v);
break;}
            case 'S':
            case 's':{cout<<"enter the
number";cin>>v.num;

      cout<<search(head,v.num)<<endl;
break;}
            case'P':
            case 'p':{cout<<"Diplay the
linked list"<<endl;
                    print(head);break;}
            }
            }
            while(c!='E'&&c!='e');
      }

      info * insert(info*head,info v){
            v.next=0;
            info* p=new info(v);
            if(head==0)
                    head=p;
            else

      if(v.num<(*head).num){
                          p-
>next=head;
                          head=p;
                    }
                    else{
                          info
*p1,*prev;
                          p1=head;

      while((p1!=0)&&(p1->num<v.num))
                          {

      prev=p1;

      p1=p1->next;
                          }
                          prev->next=p;
                          p->next=p1;
                    }
                          return head;
                    }

      info* del(info*head,info v){
            info *p,*prev;
```

khaledyassinkh@gmail.com

إعداد المهندس. خالد ياسين الشيخ

```
            if(head->num==v.num){
                p=head;
                head=head->next;
                delete(p);
            }
            else{
                p=head;
                while((p!=0)&&(p-
>num!=v.num)){
                    prev=p;
                    p=p->next;
                }

        if((p!=0)&&((*p).num==v.num)){
                    prev->next=p-
>next;
delete(p);
                }
            }
            return head;
        }
        bool search(info* head,int v){
            info* p=head;
            while(p!=NULL){
                if((*p).num==v)
                    return true;
                p=p->next;
            }
            return false;
        }
        void print(info *head){
            info *p=head;
            while(p!=NULL){
                cout<<p->num<<endl;
                p=p->next;
            }
        }
        char choice(){

            cout<<"I:inserting to linked
list"<<endl;
            cout<<"S:searching in linked
list"<<endl;
            cout<<"p:printing the linked
list"<<endl;
            cout<<"D:delete from linked
list"<<endl;
            cout<<"E:exit"<<endl;
            char c; cin>>c;
```

khaledyassinkh@gmail.com
إعداد المهندس. خالد ياسين الشيخ

```
        return c;
    }
```

باستخدام المؤشرات و بنية المكدس اكتب الإجرائيات اللازمة لتحويل عدد صحيح موجب أكبر من الصفر إلى ما يكافئه من عشري إلى ثنائي

مثال: x=8 ⟸ 1000

```
#include<iostream.h>
struct list{
    int n;
    list *next;
}*ph;
void push(list*&p,int x){
    if(x!=0){
        if(p==NULL){
            p=new list;
            p->n=x%2;
            p->next=0;
        }
        else{
            list*temp=new list;
            temp->n=x%2;
            temp->next=p;
            p=temp;
        }
        push(p,x/2);
}}
void pop(list*&p,int &res){
    if(p!=0){
        list*e=p;
        p=p->next;
        res=e->n;
        delete(e);
    }
}
void main(){
    ph=0;
    int x,res; cin>>x; push(ph,x);
    while(ph!=0){
        pop(ph,res);
        cout<<res;
    }
```

إعداد المهندس خالد ياسين الشيخ
khaledyassinkh@gmail.com

```
    cout<<endl;
}
```

اكتب برنامج بلغة c++ يقوم بالتعامل مع السلاسل(القوائم) الخطية ثنائية الاتجاه على فرض لدينا حقل واحد للمعطيات و هو الرقم num
يتضمن البرنامج الإجرائيات التالية:

1- إجرائية insert لحشر عنصر ضمن السلسلة.
2- إجرائية del لحذف عنصر من السلسلة.
3- إجرائية print_up لطباعة عناصر المصفوفة تصاعدياً
4- إجرائية print_down لطباعة عناصر المصفوفة تنازلياً.

يمكننا تخيل شكل السلسلة ثنائية الاتجاه بالذاكرة كما يلي:
كل عنصر من عناصر اللائحة يحوي الحقول التالية:

prev	num	next

حيث الحقل next هو مؤشر للعنصر التالي في السلسلة
و الحقل prev هو مؤشر للعنصر اللاحق(السابق)في السلسلة.

بداية السلسلة — نهاية السلسلة

prev	num	next	→	prev	num	next	→	prev	num	next

اختر الإجابة الصحيحة:

a. في السلسلة الخطية وحيدة الاتجاه لا يوجد مفهوم nil
b. في السلسلة الخطية ثنائية الاتجاه لا يوجد مفهوم nil
c. في السلسلة الخطية الدائرية لا يوجد مفهوم nil
d. كلمة nil تشير إلى العنوان صفر في الذاكرة
e. **c(*) و d.**

```
#include<iostream.h>
#include<conio.h>
struct info{
    int num;
    info * next,*prev;
};
info *pf=0,*plast=0;
void insert(info v){
    v.next=0;
    v.prev=0;
    info* temp=new info(v);
    if(pf==0){
        pf=temp;
        plast=temp;
    }
    else
        if((*pf).num>v.num){
            temp->next=pf;
            temp->prev=0;
            pf->prev=temp;
            pf=temp;
        }
        else
        {
            info *p1,*p2;
            p1=pf;
    while(p1!=0&&p1->num<v.num){
                p2=p1;
                p1=p1->next;
            }
            if(p1!=0){
            temp->next=p1;
            temp->prev=p2;
            p1->prev=temp;
            p2->next=temp;
```

```
                    }
                    else{
                    temp->prev=p2;
            temp->next=NULL;
                    p2->next=temp;
                    plast=p2->next;
                    }
            }
}
void del(info v){
        info *p1,*p2;
        p1=pf;
        p2=p1;
        if(pf!=0){
                if(p1==pf&& p1-
>num==v.num&&p1->next==0)
                {
                        pf=0;
                        plast=0;
                        delete(p1);
                }
                else
        if(p1==pf&&p1->num==v.num)
                        {
                                pf=pf->next;
                                pf->prev=0;
                                delete(p1);
                        }
                        else
                        {

        while(p1!=0&&p1->num<v.num)
                                {

        p2=p1;

        p1=p1->next;
                                }

        if(p1==plast&&p1->num==v.num)
                                {

        plast=p1->prev;
                                        plast-
>next=0;
                                        delete
p1;
                                }
                                else
```

khaledyassinkh@gmail.com

إعداد المهندس. خالد ياسين الشيخ

```
        if((p1!=0)&&(p1->num==v.num))
                                        {
        p2->next=p1->next;
        p1->next->prev=p2;
        delete(p1);
                                        }
                                else
        cout<<"the element is not
found"<<endl;
                        }
        }
}
bool search(int v){
        info * p=pf;
        while(p!=0){
                if(p->num==v)
                        return true;
                p=p->next;

        }
        return false;
}
void print_up(){
        info*p=pf;
        while(p!=NULL){
                cout<<p->num<<endl;
                p=p->next;
        }
}
void print_down(){
        info* p=plast;
        while(p!=0){
                cout<<p->num<<endl;
                p=p->prev;
        }
}
char choise(){
        char c;
        cout<<"I:inserting to linked"<<endl;
        cout<<"S:searching in linked"<<endl;
        cout<<"U:printing the linked
list(UP)"<<endl;
        cout<<"P:printing the linked
list(Down)"<<endl;
```

khaledyassinkh@gmail.com

إعداد المهندس. خالد ياسين الشيخ

```
        cout<<"E:exit"<<endl;
        cout<<"choise=";
        cin>>c;
        return c;
}



void main(){
        info v; char c;
        do{
                c=choise();
                switch(c){
                case'i':
                        case'I':{
                                cout<<"enter
the number"<<endl;
                                cin>>v.num;

        insert(v);break;
                                        }
                        case 'd':
                        case'D':{cout<<"enter
the number"<<endl;
                                cin>>v.num;
                                del(v);break;
                                        }
                        case 's':
                        case 'S':{cout<<"enter
the number"<<endl;
                                cin>>v.num;

        if(search(v.num))

        cout<<"found"<<endl;
                                else

        cout<<"not found"<<endl;
                                break;}
                        case 'u':
                        case
'U':{cout<<"Displaying the linked
list(UP)"<<endl;

        print_up();break;}
                        case 'p':
                        case
'P':{cout<<"displaying the linked
```

إعداد المهندس. خالد ياسين الشيخ

khaledyassinkh@gmail.com

	list(Down)"<<endl; print_down();break;} } } while(c!='E'&&c!='e'); }
اكتب برنامج بلغة c++ يقوم بالتعامل مع السلسلة الدائرية مع تحديد بنية معطيات مناسبة علما أن السلسلة لها حقلي معطيات الاسم name و الرقم num .	
كان لدى ملك ظالم طريقة مبتكرة في إعدام معارضيه إذ كان يضعهم على محيط دائرة و يبدأ العد بجهة عقارب الساعة ثم يقوم بإعدام الشخص ذو العدد n ثم يبدأ العد بعكس عقارب الساعة إلى أن يصل إلى الشخص ذو العدد n و هكذا دواليك حتى يتم تصفية جميع المعارضين و المطلوب مثل المسألة باستخدام المؤشرات مع اقتراح بنية معطيات مناسبة. ملاحظة: هذه المسألة باستخدام المؤشرات تحل باستخدام السلسلة الدائرية(هنا لا يوجد مفهوم nil)	

إعداد المهندس خالد ياسين الشيخ
khaledyassinkh@gmail.com

<table>
<tr>
<td>

```
#include<iostream.h>
#include<ctype.h>
char * f1(char *p){
        while(*p!='\0')
        {
                *p=toupper(*p);
                p++;
        }
        return p;
}
void f1(char p[],int n){
        int i=0;
        while(i<n){
                p[i]=toupper(p[i]);
                i++;}
}
void f2(char p[],int n){
        int i=0;
        while(i<n){
                if(islower(p[i]))
                p[i]=toupper(p[i]);
                i++;}
}

void main(){
        char mat[]="welcome to syria";
        cout<<mat<<endl;
        f1(mat);
        cout<<mat<<endl;
        char m[]="welcome to syria arabic";
        cout<<m<<endl;
        f1(m,23);
        cout<<m<<endl;
}
```

</td>
<td>

الخرج:

welcome to syria
WELCOME TO SYRIA
Welcome to syria arabic
WELCOME TO SYRIA ARABIC

ملاحظة: يقوم التابع toupper بأخذ محرف كوسيط ويعيد هذا الحرف الكبير الموافق له (حيث يتم تحويل الحروف من 'a' إلى 'z' إلى الحروف الكبيرة حسب ASCII) و إذا كان الحرف كبير أصلا فإنه يعيده نفسه.
و التابع toupper موجود في المكتبة ctype.h و لذلك يجب تضمينها قبل استخدام التابع.
و، لدينا التابع islower الذي يأخذ كوسيط محرف و يتحقق من كونه حرف صغير و يرد القيمة 0 في حال كان الحرف **غير** صغير و هذا التابع موجود في المكتبة ctype.h.

</td>
</tr>
<tr>
<td>

```
#include<iostream.h>
void main(){
        char mat[13]="syria arabic";
        int i=0;
        while(mat[i]!=0)
                cout<<mat[i++];

}
```

</td>
<td>

الخرج:

syria arabic

</td>
</tr>
<tr>
<td>

```
#include<iostream.h>
void main(){
        char mat[13]="syria arabic";

        while(*mat!=0)
                cout<<*mat;
}
```

</td>
<td>

الخرج:

a. طباعة syria arabic.
b. طباعة Syria فقط.
c. طباعة الحرف s فقط.
d. **(*) طباعة الحرف s بشكل لا نهائي.**
e. الجواب الصحيح يختلف عما سبق.

</td>
</tr>
</table>

إعداد المهندس خالد ياسين الشيخ
khaledyassinkh@gmail.com

```
#include<iostream.h>
void main(){
	char mat[13]="syria arabic";

	while(*mat!=0){
		cout<<*mat;
		mat++;
	}

}
```

الخرج:

a. طباعة syria arabic.
b. طباعة Syria فقط.
c. طباعة الحرف s فقط.
d. طباعة الحرف s بشكل لا نهائي.
e. **(*)الجواب الصحيح يختلف عما سبق.**

ملاحظة: البرنامج لن ينفذ لأننا نحاول تغيير اسم المصفوفة باستخدام عملية حسابية و هذا غير مقبول بلغة c++ و يمكن الاستعاضة عن ذلك باستخدام المؤشرات كالتالي مثلا.

```
#include<iostream.h>
void main(){
	char mat[13]="syria arabic";
	char * p=&mat[0];//or char * p=mat;
	while(*p!=0){
		cout<<*p++;
	}

}
```

```
#include<iostream.h>
#include<string.h>
void main(void){
	char mat[]="welcome to syria arabic";
	char a[30];
	char b[10];
	strcpy(a,mat);
	cout<<"the content in array mat is
="<<mat<<endl;
	cout<<"the content in array a   is
="<<a<<endl;
	strncpy(b,mat,9);
	b[9]=0;// or b[9]='\0';
	cout<<"the content in array b   is
="<<b<<endl;

}
```

الخرج:

```
the content in array mat is = welcome to syria arabic
the content in array a     is = welcome to syria arabic
the content in array b     is = welcome t
```

ملاحظة:

يقوم التابع strcpy بأخذ و سيطين الوسيط الأول مصفوفة الحروف و التي تكون ذات طول مناسب لتخزين به قيمة الوسيط الثاني مع الحرف الصفري و الثاني يمثل سلسلة من الحروف يتم نسخها إلى الوسيط الأول مع الحرف الصفري .

و هذا التابع موجود في المكتبة string.h لذلك يجب تضمينها قبل التعامل معه.

و التابع strncpy بنفس العمل الذي يقوم به التابع strcpy ولكن مع فارق بسيط و هو أن التابع strncpy يحدد عدد الحروف الواجب نقلها(نسخها) من الوسيط الثاني إلى الوسيط الأول و أيضاً يجب إضافة الحرف الصفري من قبل المبرمج في حال كان الوسيط الثالث في التابع strncpy أصغر أو يساوي طول السلسلة في الوسيط الثاني.

```
#include<iostream.h>
#include<string.h>
void main(void){
	char mat[20]="syria arabic";
	char *a="syria";
	char b[30]={""};//or char b[10]={'\0'};
	cout<<mat<<endl<<a<<endl;
	strcat(mat,a);
cout<<mat<<endl<<a<<endl;
strncat(b,mat,5);
cout<<"b="<<b<<endl;
strcat(b,mat);
```

الخرج:

```
syria arabic
syria
syria  arabicsyria
syria
b= syria
syriasyria  arabicsyria
```

يقوم التابع strcat بعملية إضافة للوسيط الناني الذي هو عبارة سلسلة حروف إلى الوسيط الأول الدي هو عبارة عن سلسة حروف حيث يحتل الحرف الأول من الوسيط الثاني مكان الحرف الصفري '\0' في الوسيط الأول و يجب أخذ بعين الاعتبار أن يكون الوسيط الأول الذي هو عبارة عن مصفوفة حروف أن يحوي خانات (حجر) كافية ليتم استيعابه مع

إعداد المهندس خالد ياسين الشيخ   khaledyassinkh@gmail.com

<table>
<tr><td>الوسيط الثاني كاملا بما فيه الحرف الصفري.
و يقوم التابع strncat نفس العمل إلا أنه يحدد عدد الحروف الواجب إضافتها(لصقها) من من الوسيط الثاني إلى نهاية الوسيط الأول.
و يضاف الحرف الصفري إلى السلسلة الناتجة(تلقائياً).</td><td><pre>cout<<b<<endl;

}</pre></td></tr>
<tr><td>الخرج:
1
-1
0
1
1
-1
1
-1
-1
يقوم التابع strcmp بمقارنة الوسيط الأول مع الوسيط الثاني حرفا حرفا و يعيد القيمة 0 في حال تساوي الوسيطن (السلسلتين) و يعيد قيمة سالبة في حال الوسيط الأول أصغر من الوسيط الثاني و يعيد قيمة موجبة في حال كان الوسيط(السلسلة) الأول أكبر من الوسيط الثاني.
و التابع strncmp يقوم بنفس العمل إلا أن عملية المقارنة تشمل فقط العدد المحدد من الحروف فقط و لا يقوم التابع strncmp بمقارنة الحروف التي تلي الحرف الصفري لإحدى السلسلتين.</td><td><pre>#include<iostream.h>
#include<string.h>
void main(void){
        char mat[20]="syria arabic";
        char *a="syria";
        char b[30]="hello how ara you";
        cout<<strcmp(mat,a)<<endl;
        cout<<strcmp(a,mat)<<endl;
        cout<<strncmp(mat,a,5)<<endl;
        cout<<strncmp(mat,a,6)<<endl;
        cout<<strncmp(mat,b,3)<<endl;
        cout<<strncmp(b,mat,3)<<endl;
char *m1="afd",*m2="aec";
        cout<<strncmp(m1,m2,3)<<endl;
m1="acd";
        cout<<strncmp(m1,m2,3)<<endl;
m1="a";
cout<<strncmp(m1,m2,3)<<endl;
}</pre></td></tr>
<tr><td>الخرج:
welcome
to
syria
arabic
مهمة التابع strtok هي تقسيم سلسلة حروف إلى عدة وحدات token تتألف كل وحدة من عدة حروف و يفصل بينها فواصل (قد تكون هذه الفواصل فراغات ، حروف و علامات ترقيم، حروف...
يأخذ التابع strtok وسيطين الوسيط الأول هو السلسلة المراد تقسيمها و الوسيط الثاني هو الفاصل بين وحدات السلسلة
و في البرنامج تشكل الفواصل فراغات.
و التعليمة
*p=strtok(mat," ");
تجعل المتحول p يؤشر إلى وحدة من واحدات السلسلة mat و يدل الوسيط الناني أن الفصل بين كل وحة وأخرى هي فراغ.
حيث يقوم التابع strtok بالبحث عن أول وحدة مكونة من حروف يست من حروف الفاصل (أي الفراغ) ثم يوجد الحرف الفاصل بعد أول وحدة ويضع بدا منها الحرف الصفري '\0'. مما يساعد على تحديد الوحدة التالية التي يبحث عنها التابع و يحتفظ التابع strtok بمؤشر على الحرف التالي للوحدة التالية في mat و يعيد مؤشرا على بداية الوحدة التالية نفسها
و يساعد الاستدعاء المتتالي للتابع strtokفي تحديد الوحدات التي تتألف منها mat و ذلك باستخدام القيمة NULL أو 0 أو '\0' بدلا من الوسيط الأول
حيث تفيد القيمة NULL أنه يجب على التابع strtok متابعة عمله في تحديد</td><td><pre>#include<iostream.h>
#include<string.h>
void main(){
        char mat[30]="welcome to syria
arabic";
        char *p=strtok(mat," ");
        while(p!=NULL){
                cout<<p<<endl;
                p=strtok(NULL," ");
        }
}</pre></td></tr>
</table>

إعداد المهندس خالد ياسين الشيخ
khaledyassinkh@gmail.com

<table>
<tr><td>الوحدات اعتبارا من النقطة التي وصل إليها نتيجة الاستدعاء السابق له.
و في حال عدم تبقي أي وحدة ضمن السلسلة mat يعيد القيمة 0 أو NULL.</td><td></td></tr>
<tr><td>الخرج: في حال الإدخال كان hello to syria
hello
لاحظ أن تعليمة cin تقرأ الحروف المدخلة حتى الوصول إلى فراغ أو حرف سطر جديد '\n' أو حرف نهاية ملف.
الخرج: في حال كان الدخل syriasyriasyriahowareyou

syriasyriasyriahowa
حيث تقوم التعليمة
cin>>setw(20)>>mat;
بإخبار المترجم أنه لا يجب أن تزيد عدد حروف المصفوفة عن mat عن الحد الأقصى للمصفوفة هنا يتم قراءة 19 محرف كحد أعلى و تخزن في mat و يخصص الحرف رقم ٢٠ لتخزين الرقم الصفري.</td><td>#include<iostream.h>
#include<iomanip.h>
void main(){
char mat[20];
cin>>setw(20)>>mat;
cout<<mat<<endl;

}</td></tr>
<tr><td>الخرج: في حال إدخال hello to syria
hello to syria
يسمح التابع cin.getline بإدخال سطر كامل إلى مصفوفة بما فيها الفراغات
و هذا التابع يأخذ ثلاثة وسطاء بشكل فعلي حيث يكون الوسيط الأول هي المصفوفة التي يتم تخزين السطر فيها و طول هذه المصفوفة و الحرف الذي يحدد نهاية السطر '\n' .
و ينهي التابع عمله عند قراءة الحرف '\n' أو عند الوصول إلى نهاية الملف أو عند تجاوز عدد الحروف المدخلة قيمة الوسيط الثاني منقوص منها 1 حيث يت تخصيص الحرف الأخير لتخزين الحرف الصفري '\0'
و يمكننا كتابة التعليمة
cin.getline(mat,20,'\n');
بالشكل:
cin.getline(mat,20);</td><td>#include<iostream.h>
#include<iomanip.h>
void main(){
char mat[20];
cin.getline(mat,20,'\n');
cout<<mat<<endl;

}</td></tr>
<tr><td>الخرج: في حال كان الإدخال khaled new
khaled ne
khaled ne</td><td>#include<iostream.h>
#include<iomanip.h>
void main(){
char mat[20];
cin.getline(mat,10);
cout<<mat<<endl;
char *p=mat;
while(*p!=0)
cout<<*p++;

}</td></tr>
</table>

khaledyassinkh@gmail.com

إعداد المهندس. خالد ياسين الشيخ

```
#include<iostream.h>
#include<iomanip.h>
void main(){
        char mat[20];
        cin.getline(mat,10,'d');
        cout<<mat<<endl;
        char *p=mat;
        while(*p!=0)
                cout<<*p++;

}
```

الخرج: في حال كان الإدخال khaled new

khale
khale

```
#include<iostream.h>
#include<iomanip.h>
void main(){
        char mat[20];
        cin.getline(mat,10,'d');
        cout<<mat<<endl;
        char *p=mat;
        while(*p!=0)
                cout<<*p++;

}
```

الخرج: في حال كان الإدخال syria arabic

a. syria arabic
b. syria
c. arabic
d. syria ara
e. syria arab
f. none of the above

```
#include<iostream.h>
#include<string.h>
void main(){
        char mat[30]="welcome to worcld";
        char *p=strtok(mat,"c");
        while(p!='\0'){
                cout<<p<<endl;
                p=strtok(0,"c");
        }
}
```

الخرج:

wel
ome to wor
ld

ملاحظة: الفاصل هنا بين الوحدات هو الحرف c

```
#include<iostream.h>
#include<string.h>
void main(){
char mat[30]="welcome to syria arabic";
        char *m="syria";
        cout<<strlen(mat)<<endl;
        cout<<strlen(m)<<endl;
}
```

الخرج:

23
5

ملاحظة:
التابع strlen مهمته حساب طول سلسلة حروف ما حيث يعيد عدد الحروف التي تتكون منها هذه السلسلة من دون حساب الحرف الصفري '\0'.

```
#include<iostream.h>
#include<stdio.h>
#include<string.h>
typedef char * string;
int countwithspaces(string input);
void main(void){
string mat="welcome to syria arabic";
printf("the string has %d
space\n",countwithspaces(mat));
```

الخرج:

the string has 3 space

ملاحظة: البرنامج يقوم بحساب عدد الفراغات في سلسلة حروف.

khaledyassinkh@gmail.com

إعداد المهندس. خالد ياسين الشيخ

<table>
<tr><td>

```
}
int countwithspaces(string input){
	int i,count=0;
	for(i=0;i<strlen(input);i++)
		if(input[i]==' ' )
			count++;
		return count;
}
```

</td><td></td></tr>
<tr><td>

```
#include<iostream.h>
#include<stdio.h>
#include<string.h>
typedef char * string;
static int isspace(char c);
int countwithspaces(string input);
void main(void){
string mat="welcome to syria arabic";
printf("the string has %d
space\n",countwithspaces(mat));
}
int countwithspaces(string input){
	int i,count=0;
	for(i=0;i<strlen(input);i++)
		if(isspace(input[i]) )
			count++;
		return count;
}
static int isspace(char c){
	if(c==' ')
		return 1;
	return 0;
};
```

</td><td>
الخرج:
the string has 3 space
ملاحظة: لاحظ مهمة التابع isspace
</td></tr>
<tr><td>

```
#include<iostream.h>
int digit(long n){
	int i=0;
	while(n!=0)
	{
		i++;
		n=n/10;
	}
	return i;
}
int dig(long n){
	if(n==0)
		return 0;
	else
		return 1+dig(n/10);
}
```

</td><td>
- اكتب تابع digit بلغة c++ يقوم بإعادة عدد خانات عدد صحيح لا يساوي الصفر حصرا.
- اكتب تابع ي dig عودي يقوم بنفس مهمة التابع digit.
مثال 123 يرد التابع 3
15890 يرد العدد 5
103021 يرد التابع العدد 6
-1234 يرد اتابع 4 و هكذا دواليك.....
</td></tr>
</table>

إعداد المهندس خالد ياسين الشيخ
khaledyassinkh@gmail.com

<table>
<tr>
<td></td>
<td>

```
void main(){
        long n;
        cin>>n;
        while(n==0){
                cout<<"please enter n<>0 ";
                cin>>n;
        }
        cout<<digit(n)<<endl;
        cout<<dig(n)<<endl;
}
```

</td>
</tr>
<tr>
<td>

- اكتب تابع sum عودي مهمته إعادة مجموع الأعداد الصحيحة الموجبة حصرا ضمن المجال المغلق [1..n]
- اكتب تابع sumodd عودي مهمته جمع الأعداد الصحيحة الفردية الموجبة حصرا ضمن المجال المغلق [1..n]

مثال n=15 يرد التابع sum القيمة 120
ويرد التابع sumodd القيمة 64

</td>
<td>

```
#include<iostream.h>
int sum(long n){
        if(n==0)
                return 0;
        else
                return n+sum(n-1);
}
int sumodd(long n){
        if(n==1)
                return 1;
        else
                if(n %2==1)
                return n+sumodd(n-1);
                else
                return sumodd(n-1);
}
void main(){
        long n;
                cin>>n;
                while(n<=0){
        cout<<"please enter the n>0 ";
                        cin>>n;
                }

cout<<sum(n)<<endl;
cout<<sumodd(n)<<endl;
}
```

</td>
</tr>
<tr>
<td>

اكتب تابع sumodd عودي يقوم بجمع عناصر نسق ذات الدليل الفردي حصرا
اكتب تابع sum عودي مهمته إعادة مجموع عناصر نسق.
و يطلب كتابة إجراء radmat لقراءة عناصر النسق و إجراء print لطباعة عناصر النسق.
مثال لدينا النسق التالي:

0	1	2	3	4	5	6
1	3	5	6	1	0	2

التابع sumodd يعيد القيمة 9
التابع sum يعيد القيمة 18

</td>
<td>

```
#include<iostream.h>
int sum(int *a,int n){
        if(n==0)
                return 0;
        else
                return a[n-1]+sum(a,n-1);

}
int sumodd(int *a,int n){
        if(n==0)
```

</td>
</tr>
</table>

	return 0; else if((n-1)%2==1) return a[n-1]+sumodd(a,n-1); else return sumodd(a,n-1); } void readmat(int *a,int n){ for(int i=0;i<n;i++){ cout<<"mat["<<i<<"] "; cin>>a[i]; } } void print(int a[],int n){ for(int i=0;i<n;i++) cout<<a[i]<<","; cout<<endl; } void main(){ int n; cin>>n; int * mat=new int [n]; readmat(mat,n); print(mat,n); cout<<"sum="<<sum(mat,n)<<endl; if(n!=1) cout<<"sumodd="<<sumodd(mat,n)<<endl; else cout<<"لا يوجد دليل فردي"<<endl; delete[]mat; }
اكتب إجراء max عودي لإيجاد القيمة العظمى من بين عناصر نسق. اكتب إجراء min لإيجاد أصغر عنصر في نسق.	#include<iostream.h> void max(int a[],int n,int &res){ if(n>0){ max(a,n-1,res); if(a[n]>res) res=a[n]; } } void min(int a[],int n,int &res){ if(n>0){ min(a,n-1,res); if(a[n]<res) res=a[n]; } } void readmat(int *a,int n){

khaledyassinkh@gmail.com
إعداد المهندس. خالد ياسين الشيخ

```
	for(int i=0;i<n;i++){
		cout<<"mat["<<i<<"] ";
		cin>>a[i];
	}
}
void print(int a[],int n){
	for(int i=0;i<n;i++)
		cout<<a[i]<<",";
	cout<<endl;
}
void main(){
	int n;
	cin>>n;
	int * mat=new int [n];
	readmat(mat,n);
	print(mat,n);

	int res=mat[0];
	cout<<"max=";
	max(mat,n-1,res);
	cout<<res<<endl;
	res=mat[0];
	min(mat,n-1,res);
	cout<<"min="<<res<<endl;
delete[]mat;
}
```

لدينا الطريقة التالية في التحقق من كون عدد أولي أم لا و هي كالتالي:
نحن نعلم أن أي عدد زوجي عدا 2 هو عدد غير أولي و ان أي عدد يقبل القسمة على 2 هو غير أولي.
حيث كما نعلم أن العدد الأولي لا يقبل القسمة إلا على نفسه و على الواحد مثلا تم إدخال عدد لمعرفة أن كان هذا العد أولي أم لا نبدأ بتجريب قسمة العدد على الاعداد من 2 إلى جذر العدد(دون الجزء العشري) المطلوب معرفته إن كان اولياً أم لا. فقد أثبت **علماء الرياضيات** أنه للتحقق من كون العدد اولي يكفي أن نقسم العدد على الأعداد المحصورة بين 2 و جذر العدد(دون الجزء العشري)
فإذا قبل القسمة على إحدى هذه الأعداد بلا باقي فإن هذا العدد غير أولي وإلا فهو أولي. و المطلوب كتابة تابع prime بلغة c++ يقوم بالتحقق من كون العدد أولي أم لا وفق هذه الطريقة.

```
#include<iostream.h>
#include<math.h>
bool prime(int n){
	if(n<=1)
		return 0;
	int j=(int)sqrt(n);
	for(int i=2;i<=j;i++)
		if(n%i==0)
			return false;
		return true;
}
void main(){
	int n;
	n=30;
for(int i=1;i<=n;i++)
if(prime(i))
cout<<i<<",";
}
```

الخرج: في حال n=30

2,3,5,7,11,13,17,19,23,29,

إعداد المهندس خالد ياسين الشيخ
khaledyassinkh@gmail.com

اكتب تابع sells يأخذ كوسيط قيمة المبيعات المحققة من قبل موظف المبيعات و يعيد التابع مبلغ العمولة <u>و ليس الراتب</u> المستحقة له علماً أن نسبة العمولة على الشكل التالي:
4% من قيمة المبيعات المحققة إذا كانت القيمة أقل أو يساوي من ٢٠٠٠ ليرة سورية
6% من قيمة المبيعات المحققة فيما عدا ذلك.

```
#include<iostream.h>
double sells(int n){
	if(n==0)
		return 0;
	if(n<=2000)
		return n*0.04;
	else
		return n*0.06;
}
void main(){
int n;
cin>>n;
cout<<sells(n)<<" sp"<<endl;
}
```

اكتب تابع sumdigit عودي بلغة c++ يأخذ كوسيط عدد صحيح و يرد مجموع خانات هذا العدد
مثال: n=123 يرد التابع 6.
n=-1234 يرد التابع 10-.
n=523 يرد التابع 10. و هكذا دواليك.........

```
#include<iostream.h>
int sumdigit(int n){
	if(n==0)
		return 0;
	else
			return
n%10+sumdigit(n/10);
}
void main(){
	int n;
	cin>>n;
	cout<<sumdigit(n)<<endl;
}
```

اكتب تابع عودي digiteven عودي بلغة c++ يقوم بجمع خانات العدد التي تقبل القسمة على 2 حصرا.
مثال: n=123 يرد التابع 2.
n=2241 يرد التابع 8.
n=5671 يرد التابع 6.وهكذا دواليك........

```
#include<iostream.h>
int digiteven(int n){
	if(n==0)
		return 0;
	else
		if((n%10)%2==0)
			return
n%10+digiteven(n/10);
		else
			return
digiteven(n/10);
}
void main(){
	int n;
	cin>>n;
	cout<<digiteven(n)<<endl;
}
```

إعداد المهندس خالد ياسين الشيخ
khaledyassinkh@gmail.com

البرمجة غرضية التوجه :object oriented programming(OOP)
تهدف البرمجة غرضية التوجه إلى تضمين المعطيات (الصفات attributes) و التوابع و الإجرائيات (الأفعال behaviors)أو عمليات operation ضمن مكونات تسمى الصفوف classes.
يطلق على المعطيات الخاصة بصف بالمعطيات الأعضاء data members و على التوابع الخاصة بهذا الصف بالتوابع الأعضاء member functions
أو الطرائق methods و لاستخدام هذا الصف نشتق منه أغراض objects.
إذاً يمكننا الآن أن نقول أن الصفوف تساعد على نمذجة الأغراض التي لها عدة صفات و التي يتم تمثيلها على شكل معطيات أعضاء members data كما يمكنها أن تقوم بتنفيذ عدة أفعال (behaviors) أو عمليات operations
و التي يتم تمثيلها على شكل توابع أعضاء member function
و يتم عريف الصفوف بلغة ++c بالكلمة المحجوزة class و نستخدم اسم الصف لتعريف الأغراض التي تنتمي إليه أما الطرائق methods فيتم التعامل معها من خلال عملية الاستدعاء من قبل الغرض المشتق منه.
أفضل مثال نبدأ به هو صف للتعامل مع التاريخ date.

```
#include "iostream.h"
class date{
private:int day,monthe,year;
public:
	date(int=1,int=1,int=2011);
	date(date&);
	void setday(int);
	void setmonthe(int);
	void setyear(int);
	int getday()const{return day;}
	int getmonthe()const{return monthe;}
	int getyear()const{return year;}
	void print()const;
	~date(){}
};
date::date(int d,int m,int y){
	setday(d); setmonthe(m);setyear(y);
}
date::date(date& da){
	setday(da.day+2);
setmonthe(da.monthe);
	setyear(da.year);
}
void date::setday(int d){day=d;}
void date::setmonthe(int m){monthe=m;}
void date::setyear(int y){year=y;}
void date::print()const{
cout<<day<<"-"<<monthe<<"-"<<year<<endl;
}
void main(){
	int d,m,y;
	date d1,d2(1,1,2012),d3(d1);;
//	cin>>d>>m>>y;
	d=3;
	m=5;
	y=2013;
	d2.setday(d);
	d2.setmonthe(m);
	d2.setyear(y);
	d1.print();
	d2.print();
	cout<<d2.getday()<<endl;
	d3.print();
}
```

الخرج:

```
1-1-2011
3-5-2013
3
3-1-2011
```

الشرح:
يبدأ تعريف الصف date بالكلمة المحجوزة class و يتم تحديد جسم الصف بالقوسين {}; حيث ينتهي تعريف الصف بالفاصلة المنقوطة ;
يتضمن الصف المعطيات الأعضاء التالية day, monthe, year
و التوابع الأعضاء التالية: date
,setday,setmonthe,setyear,getday,getmonthe,getyear
,print
تدعى الأسماء public و private و protected بمحددات الوصول إلى المعطيات والتوابع الأعضاء المتعلقة بالصف و تنتهي أسماء المحددات دائماً بنقطتين : مثال (:public) و ينتهي مفعول محدد ما بورود اسم محدد آخر.
و جميع المعطيات و التوابع الأعضاء الوارة بعد المحدد private لا يمكن الوصول إليها إلا من خلال التوابع الأعضاء أو التوابع الأصدقاء
و لاحظ استخدام العملية الثنائية لتحديد مجال الرؤية :: و ذلك للتوابع الأعضاء في الصف date و الواردة بعد تعريف الصف حيث يسبق اسم التابع ذكر اسم الصف متبوعاً بالعملية الثنائية لتحديد مجال الرؤية و على اعتبار أنه يمكن أن يكون لعدة توابع أعضاء من صفوف مختلفة الاسم و الترويسة نفسها لذلك تقوم العملية الثنائية لتحديد مجال لرؤية بالربط ما بين اسم التابع العضو و الصف بحيث يحصل على تحديد وحيد للتوابع الأعضاء المرتبطة بصف محدد. و إذا كان التابع inline فليس هناك ضرورة لكتابة جسم التابع خارج بنية الصف و أنما يتم ذكرها داخله و كمثال التابع getday.

إعطاء قيم ابتدائية لأغراض صف من خلال البواني :constructors

عند إنشاء غرض من صف يمكن لمعطياته أن تأخذ قيم ابتدائية بواسطة تابع يدعى بالباني مرتبط بهذا الصف و يحمل باني الصف نفس اسم الصف و لا يُحدد نمط القيمة المعادة منه و يتم استدعاؤه أتوماتيكياً عند انشاء أي غرض من الصف.
و يوجد ثلاثة أنواع من البواني :

١- الباني الافتراضي
٢- الباني ذات الوسطاء
٣- الباني الناسخ

و يتضمن مثال صف التاريخ على البواني التالية:

١- باني ذو وسطاء(وسطاء ذات قيم افتراضية)

```
date(int=1,int=1,int=2011);
```

٢- باني ناسخ

```
date(date&);
```

كيفية الوصول إلى أعضاء الصف:

تشابه العمليات المستخدمة للوصول إلى اعضاء صف تلك العمليات المستخدمة للوصول إلى حقول السجلات و بالتالي للوصول إلى اعضاء الصف باستخدام العملية (.) مصحوبة باسم الغرض أو باستخدام العملية (->) مع اسم الغرض المؤشر.

و يجب ضبط عملية الوصول إلى المعطيات و خاصة المعطيات الخاصة بشكل جيد من خلال توابع get و توابع set فعلى سبيل المثال للسماح بقراءة قيمة من المعطيات الخاصة نقوم بكتابة التابع العضو من نوع get المناسب للقيام بذلك و للسماح بتعديل قيمة من المعطيات الخاصة نقوم بكتابة التابع العضو من النوع set المناسب للقيام بهذه المهمة.

و يتضمن مثال صف التاريخ date توابع الوصول التالية:

٣- توابع الـــ get:

```
int getday()const{return day;}
        int getmonthe()const{return monthe;}
                int getyear()const{return year;}
```

٤- توابع الـــ set:

```
void date::setday(int d){day=d;}
void date::setmonthe(int m){monthe=m;}
void date::setyear(int y){year=y;}
```

و يمكننا استدعاء الطرائق methods من خلال مجموعة من الأغراض ضمن التابع الرئيسي مثلاً.

ملاحظة هامة:إذا لم يذكر اسم محدد الصفوف قبل ذكر الأعضاء فهذه الأعضاء هي private ضمناً.

الخرج:

```
99
```

```
#include<iostream.h>
class x{

        void print(){
                cout<<y<<endl;}
        int y;
public:
        x(){ y=99;}
        void f(x t){
                t.print();
        }
};
void main(){
        x t1;
        t1.f(t1);
```

khaledyassinkh@gmail.com

إعداد المهندس. خالد ياسين الشيخ

<table>
<tr>
<td>

```
}
```

</td>
<td></td>
</tr>
<tr>
<td>

```
#include<iostream.h>
class work{
public:
        int x;
        void print(){
                cout<<x<<endl;}
};
void main(void){
        work w;
        work *wp=&w;
        work &wref=w;
        w.x=99;
        w.print();
        wref.x=wref.x-9;
        w.print();
        wref.print();
        wp->x=5;
        w.print();
        wp->print();
}
```

</td>
<td>

الخرج:

99
90
90
5
5

اعد كتابة الصف work بحيث يتم تعريف العضو x على أنه عضو خاص و استخدم الباني لأعطاء قيم ابتدائية له من خلال الباني و من خلال تابع عضو في الصف work .

</td>
</tr>
<tr>
<td>

```
#include<iostream.h>
class work{
                int x;
public:
        work(int y){x=y;}

        void print(){
                cout<<x<<endl;}
        void modifity(int y){
                x=y;}
};
void main(void){
        int y=99;
        work w(y);
        work *wp=&w;
```

</td>
<td>

ملاحظة هامة:إذا لم يذكر اسم محدد الصفوف قبل ذكر الأعضاء فهذه الأعضاء هي private ضمناً.

الخرج:

99
99
99
90
5

</td>
</tr>
</table>

`` work &wref=w; y=y-9; w.print(); w.print(); wref.print(); wp->modifity(y); w.print(); wref.modifity(5); w.print(); } ```	
`` #include<iostream.h> class A{ public:A(){cout<<"A ";} ~A(){cout<<"~A";}}; class B:public A{ public:B(){cout<<"B ";} ~B(){cout<<"~B";} }; class C:public B{ public:C(){cout<<"C ";} ~C(){cout<<"~C";}}; void main(){ C c; cout<<endl;} ```	الخرج: A B C ~C~B~A
`` #include<iostream.h> class A{ public:A(){cout<<"A ";} ~A(){cout<<"~A";}}; class B:public A{ public:B(){cout<<"B ";} ~B(){cout<<"~B";} }; class C:public B{ public:C(){cout<<"C ";} ~C(){cout<<"~C";}}; void main(){ {C c;} cout<<endl;} ```	الخرج: A B C ~C~B~A
`` #include <iostream.h> class array { private: int *a; int dim,i; public: ```	الخرج: 100,300, 1,2,3,4,5,

khaledyassinkh@gmail.com
إعداد المهندس. خالد ياسين الشيخ

```
array()
{
a=new int[2];
dim=2;
i=0;
}
array(int l)
{
a=new int[l];
dim=l;
i=0;
}
void add(int d)
{
if (i>=dim)
{
cout<<"array boud exceed";
i=0;
}
a[i]=d;
i++;

}

void print()
{
for (int j=0;j<dim;j++)
cout<<a[j]<<",";
cout<<endl;
}
};
void main()
{
array a1,a2(5);
a1.add(100);
a1.add(300);
a1.print();
a2.add(1);
a2.add(2);
a2.add(3);
a2.add(4);
a2.add(5);
a2.print();
}
```

khaledyassinkh@gmail.com

إعداد المهندس خالد ياسين الشيخ

	الخرج:
`` #include "iostream" using namespace std; class index { private: int count; public: index() { count=0; } void operator++() { ++count; } void showdata() { cout<<count<<endl; } }; void main() { index c; cout<<"1 c="; c.showdata(); ++c; cout<<"2 c="; c.showdata(); ++c;; cout<<"3 c="; c.showdata(); } ```	1 c=0 2 c=1 3 c=2
`` #include<iostream.h> class test{ public:test(int=0); void print()const; private: int x; }; test::test(int value):x(value){} void test::print()const{ cout<<x<<","; cout<<this->x<<","; cout<<(*this).x<<endl; } void main(){ test t(5); t.print(); ```	الخرج: 5,5,5

khaledyassinkh@gmail.com

إعداد المهندس. خالد ياسين الشيخ

<table>
<tr><td>

```
}
```

</td><td></td></tr>
<tr><td>

```
#include<iostream.h>
class test{
public:
        test(){x=y=5;}
        int x,y;

          void print()const;
};

class op:public test{

};
void test::print()const{
        cout<<x<<" "<<y<<endl;
}
void main(){
        op t;
        t.x=t.y+t.x;
        t.y=t.x+t.y;
        t.print();
}
```

</td><td>الخرج:
10 15</td></tr>
<tr><td>

```
#include<iostream.h>
class k{
public:
        int x,y;
         void f(int n,int m){
                x=n;
                y=m;

        }
        print(){cout<<x;
        cout<<","<<y<<endl;}
};

void main(){
        k* p1=new k;
```

</td><td>الخرج:
5,10</td></tr>
</table>

إعداد المهندس. خالد ياسين الشيخ
khaledyassinkh@gmail.com

```
        k* p2=new k;
        p2=p1;
        p1->f(5,10);
        p2->print();

}
```

ملاحظة: إذا لم يذكر محدد الصفوف فهي privte ضمناً

```
#include<iostream.h>
class k{
        int x,y;
public:
         void f(int n,int m){
                x=n;
                y=m;

        }
        print(){cout<<x;
        cout<<","<<y<<endl;}
};

void main(){
        k* p1=new k;
        k* p2=new k;
        p2=p1;
        p1->f(5,10);
        p2->print();

}
```

```
#include<iostream.h>
#include<conio.h>
class t{
public:
        int x,y;
        void f(int yy){
                x=y=yy;
        }
        print();

};
t::print(){
        cout<<x<<","<<y<<endl;
}
void main(){
```

خرج البرنامج هو:

a. 9,9
b. 55,77
c. 55,55
d. 77,77
e. 77,55

khaledyassinkh@gmail.com

إعداد المهندس. خالد ياسين الشيخ

<table>
<tr><td>

```
        t r;
        t * e;
        e=&r;
        r.f(9);
        e->f(55);
        e->x=99;
        (*e).x=77;
        r.print();
        getch();
}
```

</td><td></td></tr>
<tr><td>

```
#include<iostream.h>
class aa{
private:
        int a,b;
public:

        void f(int x,int y) {a=x; b=y;}
        int f1()const {return a*b;}
};

void main(){
        aa p1;
        p1.f(10,250);
        cout<<p1.f1()<<endl;
}
```

</td><td>الخرج:
2500</td></tr>
<tr><td>

```
#include<iostream.h>
class A{
public:
        A(){cout<<"A default constructor \n";}
        A(A&){cout<<"A copy constructor
\n";}
        ~A(){cout<<"A destructor \n";}};
        class B:public A{
        public:
                B(){cout<<"B default
constructor \n";}
                B(B&){cout<<"B copy
constructor \n";}
        ~B(){cout<<"B destructor \n";}};

                class C{
                public:
                        C(){cout<<"C default
constructor \n";}
                        C(C&){cout<<"C copy
```

</td><td>الخرج:
A default constructor
B default constructor

A default constructor
B default constructor
C default constructor
B destructor
A destructor</td></tr>
</table>

khaledyassinkh@gmail.com

إعداد المهندس. خالد ياسين الشيخ

constructor \n";} ~C(){cout<<"C destructor \n";} private: B x2; }; void main(){ B b; cout<<endl; C* cp=new C;}	
#include<iostream.h> class A{ public: A(){cout<<"A default constructor \n";} A(A&){cout<<"A copy constructor \n";} ~A(){cout<<"A destructor \n";}}; class B:public A{ public: B(){cout<<"B default constructor \n";} B(B&){cout<<"B copy constructor \n";} ~B(){cout<<"B destructor \n";}}; class C{ public: C(){cout<<"C default constructor \n";} C(C&){cout<<"C copy constructor \n";} ~C(){cout<<"C destructor \n";} private: A x1; B x2; }; B fun(C *p,B b); void main(){ B b; cout<<endl; C* cp=new C; cout<<"\n before call\n"<<endl; b=fun(cp,b); }	الخرج: ملاحظة هامة الشرح التفصيلي للصفوف سيكون بالجزء الثالث.

سبحانك ربنا لا علم لنا إلا ما علمتنا

إعداد المهندس. خالد ياسين الشيخ
khaledyassinkh@gmail.com

```
B fun(C *p,B b){
        cout<<"\nin fun \n"<<endl;
        A a; cout<<endl; B bb(b);
        cout<<endl;
        delete(p);
        cout<<endl;
        return bb;
}
```

(نهاية الجزء الثانى لقاءنا فى الجزء الثالث)

"الخسارة الكبرى هى ما يموت فينا و نحن أحياء"

(اللهم صلي وسلم على سيدنا محمد و على آله وصحبه أجمعين)

ناديت باسمك يا إلهي ضارعاً إن لم تجبني فمن يجيب بكـائي
أنت الكريم فلا تدعني تائهاً فلقد عييت من البعـــــــــاد النائي
مالي سوى أعتاب جودك موئلاً فلئن رددت فمن سواك رجائي

(وما توفيقي الا بالله)



دَعِ الأَيَّامَ تَفْعَلُ مَا تَشَاءُ وَطِبْ نَفساً إذَا حَكمَ القَضَاءُ
وَلا تَجْزَعْ لِحَادِثَةِ اللَّيالي فَمَا لِحَوَادِثِ الدُّنْيَا بَقَاءُ

(من شعر الإمام الشافعي رحمه الله)

(كل ابن أنثى و إن طالت سلامتهُ يوماً على آلة حدباء محمولُ)

(إن أكبر معركة يخوضها الإنسان هي المعركة التي يغير فيها بعض عاداته و سلوكياته)